Spring 2023

Sub-Canopy Path Planning for Snow Depth Remote Sensing Using Autonomous Multi-UAVs

Debarpan Bhowmick
University of New Hampshire, Durham

Sub-Canopy Path Planning for Snow Depth Remote Sensing Using Autonomous Multi-UAVs

BY

DEBARPAN BHOWMICK
MS, UNIVERSITY OF NEW HAMPSHIRE, 2023

THESIS

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Master of Science
in
Mechanical Engineering

May, 2023

This thesis/dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering by:

Thesis Co-Director, May-Win Thein, Professor (Mechanical Engineering and Ocean Engineering)
Thesis Co-Director, Jennifer Jacobs, Professor (Civil and Environmental Engineering)
Bingxian Mu, Assistant Professor (Mechanical Engineering)

On May 15th 2023

Original approval signatures are on file with the University of New Hampshire Graduate School.

DEDICATION

I dedicate this to my parents and sister who have never doubted me and always helped me do my best. I couldn't have done this without their support and love them to the moon and back. I also dedicate this to my friends and girlfriend who have become my family and make the distance from my parents feel smaller.

ACKNOWLEDGEMENTS

Thank you to Adam Hunsaker for his countless hours behind this project. Thank you to Professor May-Win Thein and Jennifer Jacobs for their unwavering support to me. Thank you to Jacob Shaker for being my colleague and friend. And lastly thank you to the whole Advanced Controls Lab group. I am who I am because of the people around me.

This material is based upon work supported by the Broad Agency Announcement Program and the Cold Regions Research and Engineering Laboratory (ERDC-CRREL) under contract number W913E521C0006. Any opinions, findings and conclusions or recommendations in this material are those of the author(s) and do not necessarily reflect the views of the Broad Agency Announcement Program and the Cold Regions Research and Engineering Laboratory.

TABLE OF CONTENTS

DEDICATION **iv**

ACKNOWLEDGEMENTS **v**

LIST OF TABLES **ix**

LIST OF FIGURES **x**

ABSTRACT **xiii**

1 Introduction **1**

1.1 Current Snow-Depth Measurement Techniques 3

1.2 Unpiloted Aerial Vehicles 4

1.3 UAVs for Snow Depth Measurements 6

1.4 Methods 7

1.5 Thesis Contributions 8

2 Automated Path Planning **10**

2.1 A* Path Planning Algorithm 10

2.2 Assumptions and Errors 12

2.3 A* Application For Forest Surveys Using UAV 13

3 LiDAR-Based Obstacle Map Generation **15**

3.1 Study Sites 15

3.1.1 UNH Kingman Farm 15

3.1.2 UNH Thompson Farm 18

3.1.3 Ox Bow Farm 19

3.1.4 UNH Olsen Advanced Manufacturing Center 21

3.2 Above-Canopy LiDAR Scans 21

3.2.1 UAV LiDAR Systems 21

3.2.2 UAV Flight Parameters 22

3.3 LiDAR Post Processing 23

3.3.1 Generation of Obstacle Maps for A* Implementation 25

4 UAV Test Platform **32**

4.1 UAV Experimental Equipment 32

4.1.1 UAV Platform 32

4.1.2 UAV Control Components 34

4.1.3 UAV Software Modules 35

4.2 UAV Software 37

4.2.1 A* Path Planning Code 37

4.2.2 UAV Flight Control Code 38

4.2.3 Integration with A* 39

5 Preliminary Testing **41**

5.1 Initial Numerical Simulations 41

5.2 Experimental Testing Under Controlled ”Laboratory” Conditions 43

6 Field Testing of Autonomous Sub-Canopy Multi-UAV Flights **53**

7 Photogrammetry Model Results from Sub-Canopy UAV Flights 58

7.1 Sub-Canopy Flight Plan 58

7.2 Digital Surface Models (DSM) 59

7.3 3D SFM Models 60

8 Conclusions 65

8.1 Results and Suggestions 65

8.2 Future Work 66

LIST OF REFERENCES 68

A Quadcopter Dynamics and Controls 72

A.1 General Dynamics 72

A.2 General Control System 76

A.3 Inertial Frame Controllers 77

A.3.1 Inertial Position Controller 77

A.3.2 Inertial Velocity Controller 77

A.4 Body Frame Controllers 79

A.4.1 Body Frame Attitude Controller 79

A.4.2 Body Frame Angular Rate Controller 83

B Start and End Points of A* Paths 86

C Supplementary Parking Lot GPS Comparison 88

LIST OF TABLES

2.1 A* heurestic comparison 12

3.1 Tree Metrics from LiDAR 360 by Test Sites 24

4.1 Holybro S500 V2 Specifications 32

4.2 Contents of Holybro S500 V2 Development Kit 34

4.3 Flight Code Thresholds 40

B.1 UAV 1 Start and End Point Coordinates of A* Paths 86

B.2 UAV 3 Start and End Point Coordinates of A* Paths 87

B.3 UAV 3 Start and End Point Coordinates of A* Paths 87

C.1 Xsens and Pixhawk 4 GPS Measurement Comparison Statistics 88

LIST OF FIGURES

1.1 a) Spatial resolution vs study site size, b) Cost of operation per square meter vs size of study site [1] 5
1.2 UAV Platforms 6

2.1 A* Visualization [2] 11

3.1 Kingman Farm Located in Madbury, New Hampshire 17
3.2 Photos of The Thompson Farm Forest Site Located in Dover, New Hampshire 19
3.3 Photos of the Ox Bow Farm in Newmarket, NH 20
3.4 DJI Matrice 300 23
3.5 3D Point Cloud and Subsequent Altitude Band Map For Three Test Sites . . 27
3.6 Obstacle Percentage by 1m Altitude Bands and Average Tree Heights for Three Test Sites 28
3.7 Altitude Map Comparison for Three Sites 30
3.8 Kingman Farm 2-3 m Altitude Band Map Showing Open Space and Obstructed Space 31

4.1 Holybro S500 V2 UAV 33
4.2 Holybro S500 V2 UAV 33
4.3 Rasberry Pi integrated with Holybro S500 35
4.4 ROS and PX4 Communication Architecture [3] 36

5.1 Gazebo Simulation Planned and Actual Path for Kingman Farm 42

5.2 UNH Olsen Advanced Manufacturing Center Test Site (Red Dot Showing Parking Lot) 43

5.3 Xsens Mti g710 GPS (Blue Circle) and Pixhawk 4 GPS (Red Circle) 44

5.4 Test 1 at UNH Olsen Advanced Manufacturing Center Parking Lot Comparing Xsens (Orange) and Navigational GPS Measurements (Blue) 45

5.5 2-3 m Altitude Band Map of Kingman Farm 47

5.6 5-6 m Altitude Band Map of Kingman Farm 48

5.7 7-8 m Altitude Band Map of Kingman Farm 49

5.8 9-10 m Altitude Band Map of Kingman Farm 50

5.9 2-10 m Altitude Band Map of Kingman Farm 51

5.10 Pre-convolution and Post-Convolution 2-10 m Altitude Band Maps 52

6.1 Kingman Farm Snow conditions During January to March 2023 54

6.2 Actual and Planned missions for the three UAVs Sub-Canopy in the Kingman Farm 2-10m Altitude Band Map (March 2023). Obstructions are Trees . . . 55

7.1 DSM Comparing Photogrammetry Results: (a) Below-Canopy and (b) Above-Canopy 59

7.2 Above-Canopy 3D SfM RGB Model 61

7.3 Sub-Canopy 3D SfM RGB Model 62

7.4 Altitude Profiles of SFM Models: Sub-Canopy Model (Top) and Above-Canopy Model (Bottom)) 63

7.5 Above-Canopy UAV View (left) and Sub-Canopy UAV View (right) of the Same Area 63

A.1 Quadrotor dynamics 73

A.2 Overall Px4 Control Scheme 76

A.3 Position Px4 Control Scheme 77

A.4 Velocity Px4 Control Scheme 78

A.5 attitude Px4 Control Scheme 79

A.6 attitude rate Px4 Control Scheme 83

A.7 Standard Attitude Rate Px4 Control Scheme 84

A.8 Parallel Attitude Rate Px4 Control Scheme 85

ABSTRACT

Sub-Canopy Path Planning for Snow Depth Remote Sensing Using Autonomous Multi-UAVs

by

Debarpan Bhowmick
University of New Hampshire, May, 2023

NASA studies indicate that 68% of the Earth's fresh water exists in the form of snow and ice. As such, analyzing global snow fall patterns is a useful tool with which scientists can extract the quantity of fresh water present in both the atmosphere and on the ground at any given time. The goal of this research is to leverage autonomous Unpiloted Aerial Vehicles (UAVs) to measure snow depth on the forest floor via sub-canopy flight. To enable such remote sensing missions, overhead Light Detection And Ranging (LiDAR) scans are used to aid in pre-determined UAV flight path planning. This results in autonomous sub-canopy missions that are able to avoid obstacles (e.g., trees, branches, and flora) and provide optimal LiDAR-based snow depth measurements.

The A-star (A*) algorithm is the chosen path planning method for this research and is used to determine appropriate flight plans for multi-UAV missions.Ox Bow Farm, Kingman Farm, and Thompson Farm are evaluated and have sub-canopy tree density between 60-90%, 20-42% and 25-55% respectively. Proof-of-concept testing is performed at the University of

New Hampshire Kingman Farm in Madbury, New Hampshire. Field tests show that this method is viable for under-canopy snow depth measurements when tree density is below 20%. In addition to the added efficiency of an autonomous multi-UAV mission (as opposed to a single, remotely operated UAVs), the resulting sub-canopy photogrammetry results, from which snow depth measurements can be extracted, are shown to provide improved ability to capture snow as compared to above-canopy flights.

CHAPTER 1

Introduction

Measuring global snow fall is critical to understanding the planet's ever-changing climate. "Snow cover" refers to the total amount of land covered by snow at any point in time. This metric is affected by the amount of snow that is allowed to reach the ground, the air temperature and the rate at which snow ablation occurs [4]. Snow cover has a mutual relationship with its environment and climate. Snow is a highly reflective material because of its white color. Fresh snow absorbs 10-20% of the sunlight that reaches the surface, whereas bare ground absorbs 70-90% of the radiation. [5] [6] [7]. Therefore, greater snow cover leads to the cooling of the environment, while less snow cover results in a heating effect. The air temperature influences the amount of snow that falls to the ground. In this way, snow cover both affects and is affected by the climate. On a smaller scale, snow is essential in providing insulating effects on which many plants and animals rely to survive frigid winter temperatures. Humans rely on snow to replenish rivers, lakes, and groundwater for drinking water. The soil moisture is also influenced by snow. When snow melts earlier in the season, the soil tends to dry quicker which can, in turn, lead to droughts and increase the risk of wildfires.

While precipitation occurs as snow in many parts of the world, snow in forested regions is particularly difficult to measure. Due to difficulties in traversing the snow terrain, snow measurements are often challenging to obtain even in open areas. However, in forested areas, snow data collection becomes a more arduous task because trees and other obstacles hinder the rate and accuracy of the data. Approximately one-fifth of the land in North America is covered by forests [8], and the snowfall accumulating in these regions cannot be neglected when trying to characterize a water and energy balance.

Some challenges to measuring snow depth in forests include access, complete coverage of the area of interest, and quick retrieval of data to make sure it is relevant and up-to-date. Data collected in these regions is often sparse and thus, results are incomplete [9]. To improve the accuracy of snow depth in forests, new strategies are needed to increase the reliability and confidence in data collection methods.

Autonomous and fast-data collection may be a cost-effective method. Unpiloted Aerial Vehicles (UAVs) are agile platforms that can be fitted with sensors and can be used to collect data in forests. Current UAV-based snow detection methods rely on over-canopy flights to measure snow depth [10]. Light Detection And Ranging (LiDAR) sensors are fitted to the underside of UAV platforms and flown over the canopy of interest. The LiDAR sensor sends out laser pulses and times the return of the beams. This allows the sensor to build a three-dimensional (3D) model of the objects from the reflected pulses. A snow-on flight is conducted to construct a Digital Elevation Model (DEM) of the forest floor. The DEM is classified and normalized according to the points at which the pulses return from the forest floor. After snow melt occurs, a snow-off flight is conducted to create a new DEM. The ground points of the snow-on and snow-off flights are subtracted to calculate the snow depth on the forest floor. However trees and branches obstruct the LiDAR pulses, so there are areas of the forest floor where no data can be retrieved. This leaves discrepancies in the snow depth measurements. Therefore, bringing the LiDAR sensor closer to the forest floor will alleviate many of these problems. A method is necessary to allow the UAV to safely navigate under the canopy of the forest. This would remove the obstructions for the LiDAR pulses and allow for a complete DEM to be constructed for the snow depth measurements.

Real-time detection of these obstacles is difficult, time consuming, and costly. A priori knowledge of the environment (e.g, the presence of any obstacles) can ease UAV computation sensor payloads. Above-canopy LiDAR maps can be leveraged to build this knowledge base from which a safe path can be planned to collect snow depth data in sub-canopy regions.

The goal of this research is to present the methods used to map below-canopy snow depth by flying UAVs below the forest canopy. Furthermore, the basis of path planning and navigation is discussed. Simulations of UAV navigation are conducted. A final real-world test is performed to evaluate the viability of the method.

1.1 Current Snow-Depth Measurement Techniques

Typically, snow depth measurements on forest floors are collected using manual instruments, such as snow tubes that measure snow depth at a fixed point within the forest. These methods require transporting a significant amount of equipment to inaccessible locations. In fact, sampling an entire forest may not be possible due to lack of accessibility [11]. These challenges result in measurements of only easily accessible areas, which are notably smaller in number, and result in less reliable data. A model can also be used to extrapolate snow depth data to larger regions [12] [13] [14]. Satellite-based data collection methods automate snow observations [15] [16]. However, these methods are less accurate, expensive, and time-consuming.

Satellite-based remote sensing has a long history of forest health monitoring and offers advantages but has its own disadvantages [1]. Satellite swaths are large enough that they can gather data on vast expanses of forest-covered land. Nevertheless, satellite-based remote sensing may lack the spatial and temporal resolution necessary to observe some forest metrics [17]. This is particularly important when monitoring snow conditions because changes occur rapidly in forests, and snow conditions have high spatial variability [18] [19]. Satellite overhead passes may occur only weekly or even less frequently for specific forest sites and cannot measure snow for dense forest canopies. Due to these factors, satellites cannot achieve the spatial resolution required to accurately capture snow in forests.

Another method of data collection uses piloted aircraft carrying remote sensing payloads. Piloted aircraft are able to measure forest metrics with high spacial resolution because of their proximity to the forest floor [20] [21]. Logistical, financial and safety issues, however, make piloted aircraft flights difficult. The costs are very high making this method viable only for large areas of data collection [22]. The resolution of piloted flight data can be very accurate but the challenges in obtaining piloted flight data make it unviable in many situations.

1.2 Unpiloted Aerial Vehicles

UAVs have emerged as a popular platform for many remote sensing applications [23]. The quadrotor UAV is a small versatile platform which can carry various sensor packages. Sensor innovation has led to smaller form factors [24]. The combination of small aerial vehicles and small sensor packages allows users to apply these technologies to remote sensing applications such as wildfire detection, forest health monitoring, and forest preservation. While UAVs cannot cover as large an area as that for satellites and piloted aircraft, UAV's are seldom matched in their spatial resolution, cost efficiency, ease of operation, and repeatability of data collection.

Figure 1.1 shows how typical piloted aircraft and UAV's compare in terms of resolution and cost in different-sized study sites. While UAVs and piloted aircraft offer similar resolutions, piloted aircraft are capable of surveying larger areas as seen in 1.1a. However, UAVs have a much lower cost-to-area ratio as seen in 1.1b.

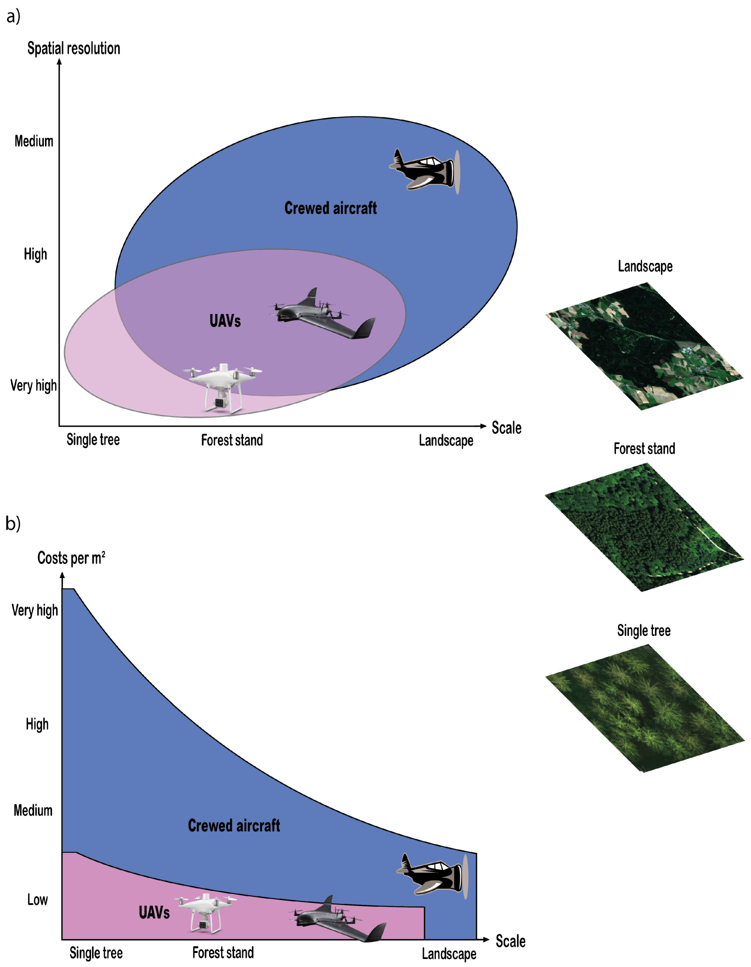
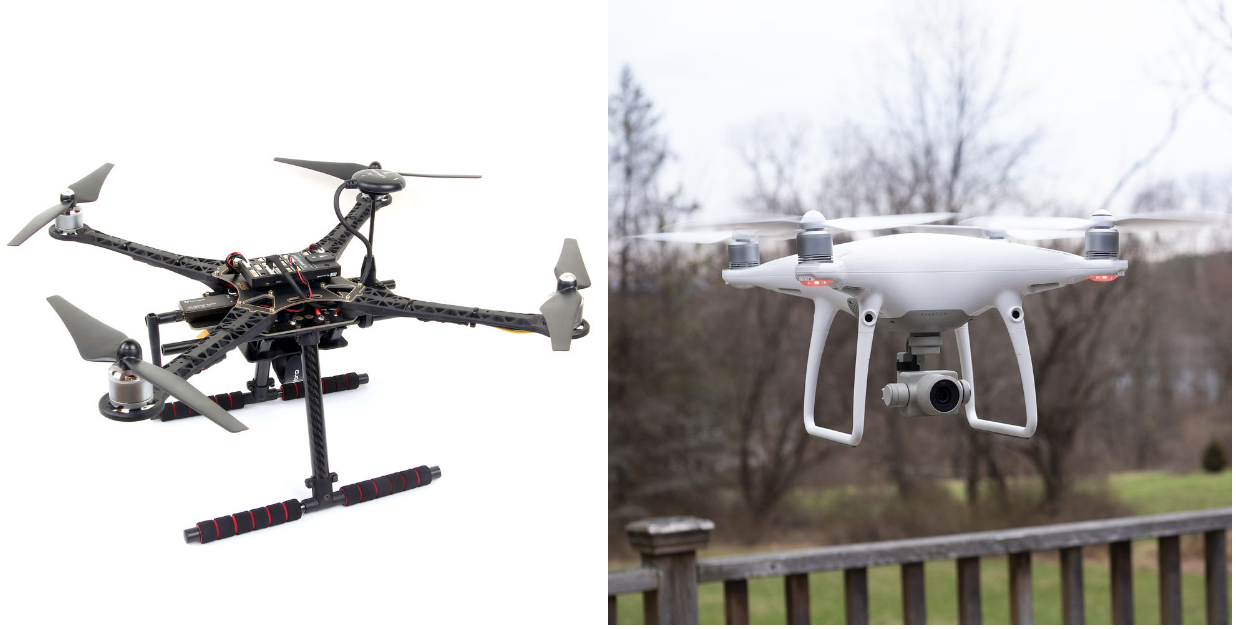


Figure 1.1: a) Spatial resolution vs study site size, b) Cost of operation per square meter vs size of study site [1]

In recent years, researchers have been trying to utilize UAVs to collect snow depth data [25]. Most of these efforts involve flying over the forest canopy using optical LiDAR sensors that penetrate past the canopy to extract information regarding the snowpack below. This method has been somewhat successful [10], but trees and branches obstruct the LiDAR pulses. Therefore, a method which allows the UAV to travel within the canopy would allow for improved resolution on snow depth measurements.

1.3 UAVs for Snow Depth Measurements

Based on cost and effective data collection, UAVs are a robust platform for remote sensing applications in forest snowpack monitoring. As such, this thesis proposes to use a quadrotor UAV platform for snow depth remote sensing. UAVs are cost-efficient, making this technology widely accessible, and it can be manufactured with a small form factor which makes it ideal for performing below canopy flights. Quadrotor UAVs consist of four arms attached to a central platform with rotor blades at the end of the arms. These platforms can execute movements in six degrees of freedom and are agile and have fast time responses. The two quadrotor UAVs used in this research are the Holybro S500 and the DJI Phantom 4. The Holybro S500 (Figure 1.2) is used to validate the navigation tracking.



(a) Holybro S500 [26]

(b) DJI Phantom 4 [27]

Figure 1.2: UAV Platforms

1.4 Methods

UAVs equipped with high resolution 3D LiDARs are flown at 20, 30 and 40 m above the forest canopy in a lawnmower path to collect dense point cloud data. The data from the different flights are merged to build a 3D model of the forest environment. These models show the layout of the forest from the forest floor to the canopy crown.

The 3D models are processed to create a series of two-dimensional (2D) altitude models. Each 2D map characterizes the forest at one altitude. The ratio of free space to obstructed space at each altitude band is calculated to determine the least densely packed altitude band below the canopy. To produce these maps, LiDAR returns that lie within an altitude range are extracted.

The 2D altitude map of LiDAR returns are gridded at a 1x1 m pixel resolution. The A-star (A*) path planning algorithm is applied to the maps to find an efficient obstacle-free path between adjacent points throughout the forest. This process is repeated between multiple points under the canopy to produce a full-coverage mission plan. These paths can be used to autonomously navigate UAVs for LiDAR-based and photogrammetry-based snowpack data collection. For this research, pilot activities are conducted using high resolution 4K cameras. Structure for motion algorithms are used to construct 3D point clouds from frames of the video recorded during under-canopy flight.

The autonomous path planning algorithm used in this research is the A* method. A* is a path finding algorithm which has become a standard in the research community and industry [28]. A* based path planning is a popular method for UAV navigation for low altitude flights [29]. The search algorithm is straightforward to implement and ensures optimality. Grid-based search areas also work well with the algorithm. For these reasons, A* path finding is selected to be employed on the LiDAR-augmented maps of the search area.

This thesis proposes the use of multiple UAVs to autonomously navigate under the forest canopy to collect snow depth data. Autonomous UAV technology has vastly improved as more industries and people realize their potential to further remote sensing applications. Yet, one of the areas where there is still limited autonomous navigation capacity is UAV flight planning in forests. The technologies for within-canopy navigation are very expensive, complicated and not open source [30]. To make forest snowpack data widely available, cost-effective data collection is necessary. In this study, the A* path planning method is used on high-resolution LiDAR maps of the forest to pre-plan paths for the UAV platforms. Here a Holybro S500 UAV platform is used with a Raspberry Pi microprocessor to plan and execute a flight path within the forest canopy. The fully built platform is constructed for less than $1000 (minus the sensor payload), a significant cost reduction compared to that used in current state-of-the-art technology.

The method of sub-canopy, pre-planned autonomous multi-UAV navigation is first tested and confirmed in Gazebo, a numerical simulation software, to accurately model UAV dynamics and the hardware and software used to command it. An empty parking lot in Durham, New Hampshire is used to test the Holybro S500 UAV platforms. These tests are used to determine the levels of GPS error in the navigation system so that appropriate safety buffers may be added to obstacles in the actual forest environment. A 0.02 km^2 forest in Madbury, New Hampshire is selected as the field test site, where three coverage paths are planned. Here, three UAVs are flown concurrently to demonstrate the viability of using multi-UAV autonomous navigation under forest canopy for snow-on and snow-off data collection.

1.5 Thesis Contributions

The contributions in this work are as follows:

- A method to perform LiDAR based mapping of forest environments is developed, which leverages over-canopy UAV flights.

- Altitude maps of sub-canopy environments are produced, which accurately detect and localize any obstacles present.

- The A* path planning is implemented using altitude maps for safe navigation of multiple UAVs in sub-canopy regions.

- Digital Elevation Models are produced from Structure From Motion methods by flying a single UAV under the forest canopy and are shown to be both dense and accurate.

The thesis is organized as follows, Chapter 2 discusses automated path planning techniques, particularly the A* method. Chapter 3 describes the test site used in this study, as well as the methods use to obtain appropriate UAV LiDAR scans from which A* automated path planning is determined. The UAV test platform is described in Chapter 4. Chapter 5 describes numerical simulations and "Laboratory" experiments to test UAV systems. Chapter 6 discusses the results of sub-canopy path planning methods in a forest test site. The resulting photogrammetry model from the autonomous multi-UAV missions in this research is presented in Chapter 7. Chapter 8 discusses conclusions and future work.

CHAPTER 2

Automated Path Planning

2.1 A* Path Planning Algorithm

The A* path searching algorithm is an industry standard and is used in many applications such as automated factory lines, automated seaport transport vehicles and many fields of computer science such as file path locating and the video game industry [31]. It is a heuristic-based path search algorithm which guarantees an optimal-path solution (if one exists) between any two given points in a search space. A* uses a heuristic cost function which is minimized based on the start area and its distance away from the desired end point. The heuristic cost function is represented by the following equation:

$$f\left(n\right) = g\left(n\right) + h\left(n\right) \tag{2.1}$$

where $g(n)$ is the cost or distance of the start node to any node in the search space, and $h(n)$ is the distance from any arbitrary node to the desired end. At each step of the algorithm's execution, this function is minimized reducing the overall cost associated in the path that is being produced, ensuring optimality. The algorithm begins by extending a path from a first node to a second node which will result in the lowest cost. This method is repeated for each iteration until the end node is reached. The cost function $f(n)$ may be chosen to weight any cost according to distance, time, or other variable specific to the search problem of interest. If the function is chosen such that it never underestimates the actual cost to get the goal, A* is guaranteed to find the least costly path. Most implementations of this algorithm use what is called a "priority queue" to analyze and store the nodes of interest at each step. In this queue, nodes that have the least cost (i.e, smallest value of $f(n)$ are prioritized to be removed until the removed node is found to be the goal node. In Figure 2.1, the h score of the nodes A and B are 4 and 5 respectively. Therefore traveling from S to A is less costly, but the G of travelling from B to E is less expensive than the cost of A to E. Therefore the most efficient path is S to B to E even though the initial S to A is costlier than S to B.

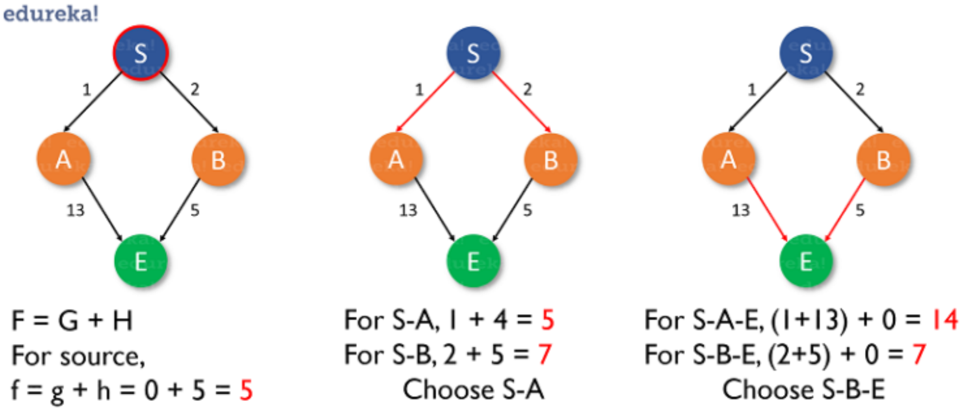


Figure 2.1: A* Visualization [2]

In problems where the search space is a physical environment and the path being produced is traversed through space, the $g(n)$ and $h(n)$ are chosen to be a type of distance calculation. The three most popular choices for the heuristic are Euclidian distance (Equation 2.2), Manhattan distance (Equation 2.3), and diagonal distance (Equation 2.4) [28]. In most square-grid cases, which have eight directions of movement, the Euclidian distance is chosen because it rarely overestimates $h(n)$. Therefore, the optimal path is always found at the cost of computation time.

The choice of the heuristic function is integral to the behavior of the algorithm and can be manipulated based on the specific search problem. $h(n)$ and $g(n)$ are balancing terms and estimating these values accurately allows A* to perform faster (or slower) and find shorter (or longer) paths (Table 2.1). When $h(n)$ is estimated to be lower than the actual cost of moving from the current node to the goal, A* is guaranteed to find the path with the lowest cost. But the algorithm will take longer to execute because it will expand and analyze more nodes. If $h(n)$ is estimated to be exactly equal to the cost of moving from the current node to the goal, A* will provide the most efficient path and not expand to any node outside of the best path. When $h(n)$ is estimated to be higher than the actual cost of moving from the current node to goal node, it is not guaranteed to return the best path, but it will shorten computation time.

Estimated $h(n)$	A* performance
$<h(n)$	Finds best path but requires more computation time
=h(n)	Finds best path in the least amount of time
$>h(n)$	Finds a non optimal path but with less computation time

Table 2.1: A* heurestic comparison

2.2 Assumptions and Errors

In this work, the forest is assumed to be a static environment. Because the UAV travels above the forest floor but below the canopy, this air space can be assumed to be constant (i.e, That is no small plants and animals are present.) Tree trunks and branches are the main obstacles in this space. These objects are fairly resistant to most biological and weather factors (e.g, wind and snow) and is assumed to alter the tree trunks insignificantly. However, inclement weather can alter these large structures, and caution should be used by users when such events take place.

It is assumed that the canopy cover is such that Global Positioning System (GPS) are still available in the sub-canopy environment. In deciduous forests, the canopy cover decreases during the wintertime as many trees lose their foliage. Therefore, the largest source of error for this research lies in the accuracy of the GPS signal. Foliage cover can obstruct GPS signals and decrease the accuracy of the UAV localization coordinates. Other sensors on the UAV (e.g, gyroscope, accelerometer, and magnetometer) are sensed locally and do not rely on satellite-based signals. Virtual buffers may be added to the obstacles to take into account degraded GPS signals to be used in the under-canopy environments.

2.3 A* Application For Forest Surveys Using UAV

2D altitude obstruction maps are represented as matrices of 1 and 0 (where 1 represents obstacle and 0 represents free space). This allows for direct application of the A* path finding algorithm to plan paths for UAV navigation. When the start node and end node is specified in the search space, A* produces a series of pixels which lead from the start node to the goal node and can be used by the UAV for sub-canopy navigation. Because path planning is completed offline, computational time is prioritized lower than the required mission distance traveled. The heuristic chosen for this applicant is the Euclidian distance (Equation 2.2) to ensure optimal path production at the cost of computation time. Additionally, Euclidian distance is the simplest and most computationally-efficient method in comparison to other heuristic measures.

Pixels with dimensions of 1 m by 1 m are used to offer adequate pixel resolution to maneuver the UAV. While larger UAV platforms require larger pixels, denser point cloud retrieval allows for smaller pixel sizes, providing greater UAV maneuverability. Obviously dense forest canopies require smaller pixel sizes to discern between obstacles and free space.

The Equation for Euclidan distance is:

$$d(x,y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2.2}$$

The Equation for Manhattan distance is:

$$d(x, y) = \ |x_1 - x_2| + |y_1 - y_2| \tag{2.3}$$

The Equation for diagonal distance is:

$$d(x, y) = \ max(|x_1 - x_2| \, , |y_1 - y_2|) \tag{2.4}$$

CHAPTER 3

LiDAR-Based Obstacle Map Generation

There are many ways to autonomously navigate a robotic agent in different environments. In most cases, a priori knowledge of the environment is required because of the computational burden of real-time localization and planning. Errors in sensor feedback can also cause such methods to fail and result in a catastrophic systems failure. To alleviate these issues for autonomous sub-canopy navigation, a reliable method of collecting environmental information is required. Three different study sites are evaluated in order to understand the limitations of sub-canopy obstacle density with regards to autonomous flight navigation. This chapter discusses the study sites where physical tests are conducted in order to determine appropriate methods for obtaining obstacle maps and viable environments for A* implementation.

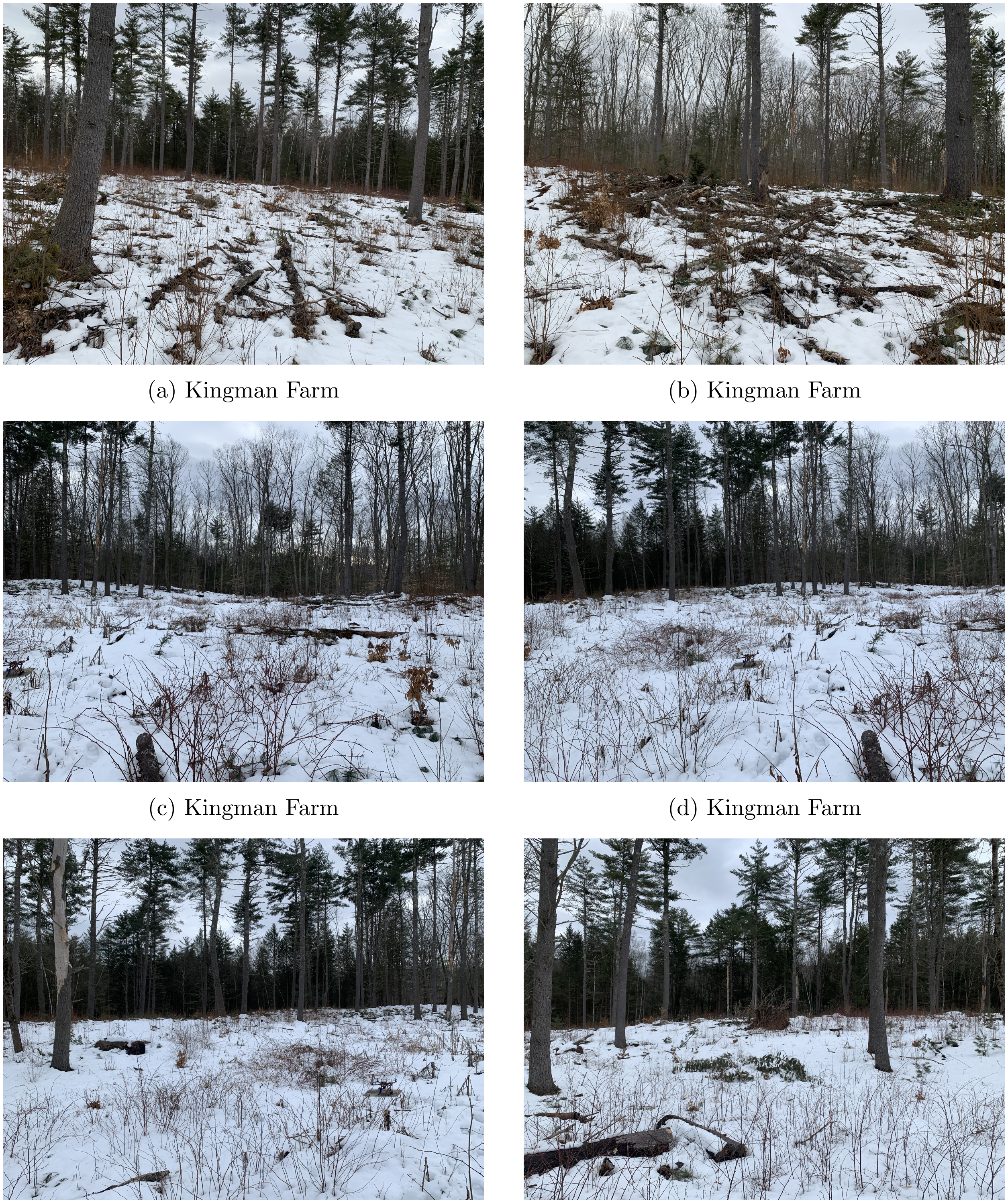
3.1 Study Sites

3.1.1 UNH Kingman Farm

The first study site is a forest located in Madbury, New Hampshire. The site is located just south of the Bellamy River and is a part of the University of New Hampshire (UNH) Kingman Farm property (43.1743058 latitude, -70.9308772 longitude) (Figure 3.2). The site is approximately 0.02 km^2 in total area. The mature trees in the forest are 100 percent white pine trees. The average diameter of the tree trunks is approximately 0.5 m measured at a height of 1.4 m above the forest floor. The basal area, which describes the proportion of the study site occupied by tree trunks, is calculated to be 1263 square meters per square kilometer (personal communication with Stephen Eisenhaure).

Based on the findings, approximately 75% of the ground area in the understory of the surveyed region is occupied by woody plants. The remaining growing space on the forest floor is occupied by grasses and other herbaceous material. Among the woody plants, black birch holds the largest growing space in the range of 0.9-1.5 m, although other species (e.g, rubus spp, American beech, red oak, hemlock, white pine, and red maple) are also present (in descending order of abundance). In the smaller size class of 0-0.9 m, particularly in the size range of 0-0.3 m, white pine is the dominant species. Other species, however, occur in roughly equal abundance in the larger size classes. Although individual stems of all species can be found occasionally in the 1.8-2.7 m size class, these are rare and represent the tallest individuals of their age group. (Pictures of the test sire are provided in Figure 3.1.).

From these observations, the ground is seen to be very dense up to an altitude of 1.5 m. Above this, the airspace is clear aside from the white pine trees. White pine tree trunks typically have a straight and cylindrical shape which allows for a uniform and static environment.



(a) Kingman Farm

(b) Kingman Farm

(c) Kingman Farm

(d) Kingman Farm

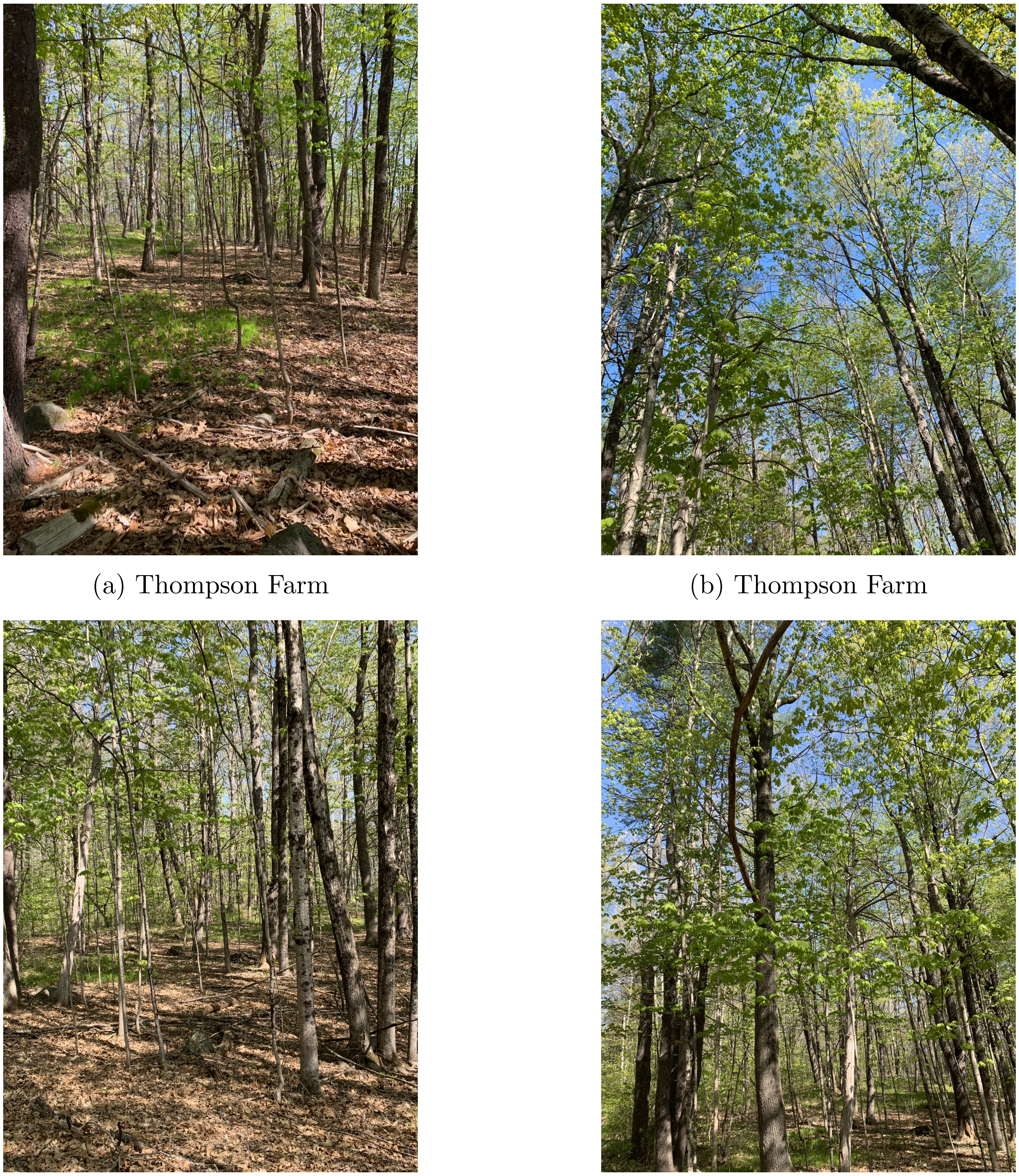
(e) Kingman Farm

(f) Kingman Farme

Figure 3.1: Kingman Farm Located in Madbury, New Hampshire

3.1.2 UNH Thompson Farm

The second study site is located at the UNH Thompson Farm Research Observatory in southeastern, New Hampshire (43.10892 latitude, -70.94853 longitude) (Figure 3.2). The site consists of hardwood forests and open fields. Thompson Farm covers an area of 0.83 km^2 and has minimal variation in land elevation [32]. The forested areas mainly consist of white pine, northern red oak, red maple, shagbark hickory and white oak.



(a) Thompson Farm

(b) Thompson Farm

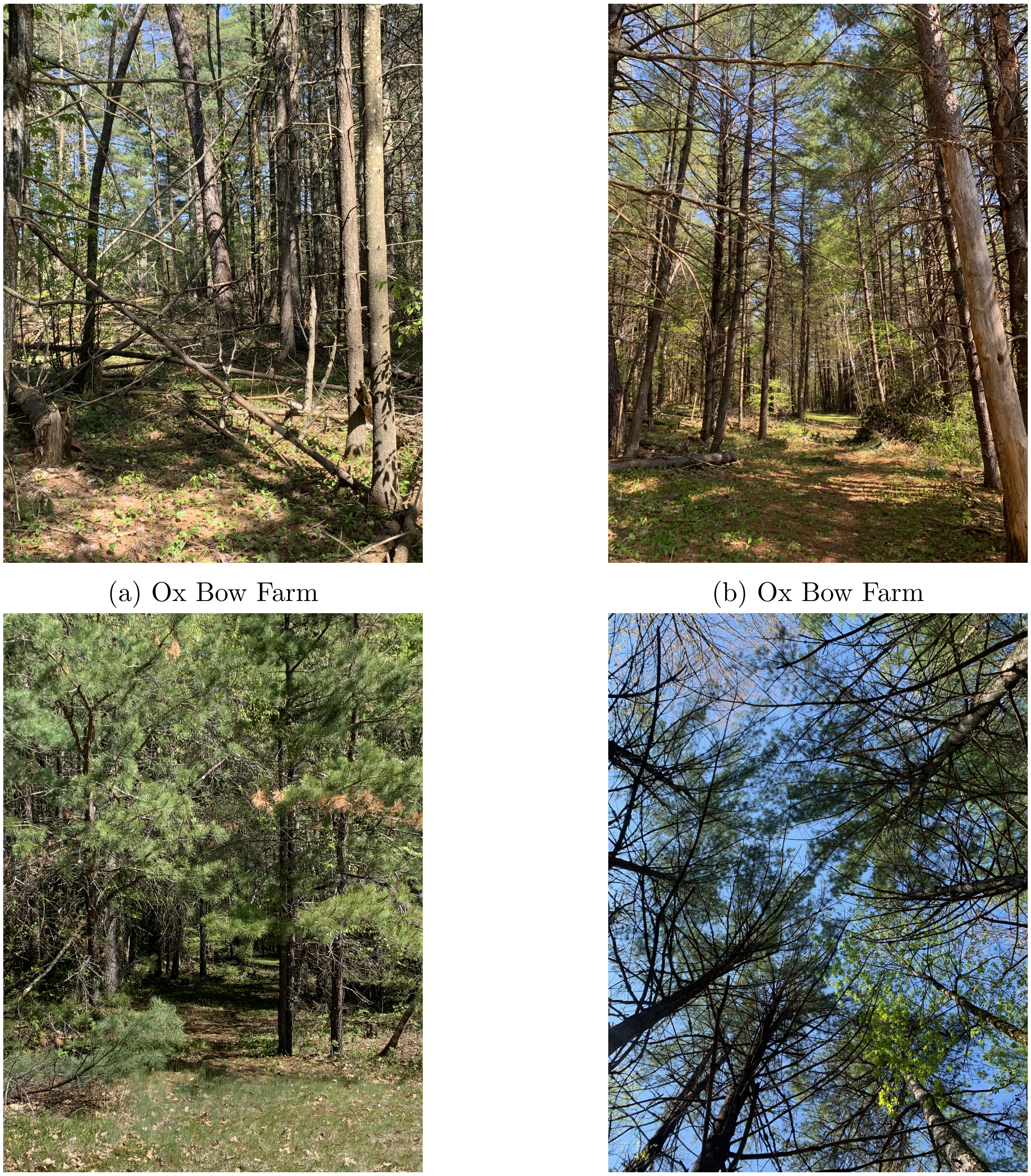
(c) Thompson Farm

(d) Thompson Farm

Figure 3.2: Photos of The Thompson Farm Forest Site Located in Dover, New Hampshire

3.1.3 Ox Bow Farm

The third study site is the Ox Bow Farm property located in Newmarket, New Hampshire. Ox Bow Farm covers an area approximately 0.016 km^2. The site is considered to be an upland pine forest (personal communication Jennifer Jacobs). The main type of trees present are white and pitch pine. The forest has a dense sub-canopy and considerable tree-fall.



(a) Ox Bow Farm

(b) Ox Bow Farm

(c) Ox Bow Farm

(d) Ox Bow Farm

Figure 3.3: Photos of the Ox Bow Farm in Newmarket, NH

3.1.4 UNH Olsen Advanced Manufacturing Center

A secondary, open test site is used to validate flight dynamics. This site is in Durham, New Hampshire (43.154575, -70.957095). The location is the parking lot of the UNH Olsen Advanced Manufacturing Center, which measures 6143 m^2. This site is used to provide controlled, laboratory conditions.

3.2 Above-Canopy LiDAR Scans

Above-canopy UAV LiDAR surveys are conducted to generate obstacle maps of the area of interest. A LiDAR system is rigidly mounted to a quadcopter and a lawnmower flight path is performed over the canopy. The resulting 3D point cloud collected is decomposed into 2D obstacle maps that describe the sub-canopy environment.

3.2.1 UAV LiDAR Systems

LiDAR mapping is conducted using a compact GreenValley LiAir V70 LiDAR system with dimensions of 110 mm x 81.6 mm x 140.2 mm and weighing 1.1 kg. The LiAir V70 uses a Livox AVIA laser sensor and has a measurement range of 190 m at 10% reflectance and 450 m at 80% reflectance. The system can collect 240,000 points per second at an accuracy of ±5 cm. The sensor has a field of view of 70.4° horizontally and 4.5° vertically which allows for complete 3D point clouds of the forest to be produced for overhead missions. The Livox sensor releases laser pulses in an irregular scan pattern to reduce systematic distortions in data. The length between the emission source on the sensor to the ground is calculated using the Time Of Flight (TOF) principle. Each pulse that the Livox emits can receive up to three reflected signals. This allows the sensor to penetrate past obstructions and discern them from the forest floor. For example, the leaves and branches block the forest floor from the position of the UAV. Receiving three reflected signals enables the sensor to correctly sample the vertical profile of the canopy with the last reflected signal likely representing the forest floor.

The LiAir combines a laser scanner along with a Global Navigation Satellite System (GNSS) receiver and Inertial Measurement Unit (IMU) readings to create a georeferenced feature map. The georeferencing systems have an altitude accuracy of 0.008° (±1 standard deviation) and Azimuth accuracy of 0.038° (±1 standard deviation). The measurement rates of GNSS and IMU are such that each point from the laser return is matched with a UAV orientation and location return. The onboard GNSS measurements are differentially corrected for each flight with the aid of a fixed base station that records the necessary data. LiGeoreference software allows the user to correct the recorded data by taking into account the average recorded position of the base station.

3.2.2 UAV Flight Parameters

A DJI Matrice 300 UAV (shown in Figure 3.2) is equipped with the LiAir, and an above-canopy lawnmower-shaped coverage over the test site is programmed into the UAV. Flight speed is 2 m/s or less to ensure a dense point cloud (Sullivan et al.). At each site, three flights are performed at altitudes of 30 m, 40 m, and 50 m above the forest floor during the survey. All flight patterns are identical. Each consisted of a lateral lawnmower path with six lines and longitudinal lawnmower path with six lines with each line separated by 20 m. Repeated flights produce a high resolution point cloud that is able to captured fine-scale obstacles found underneath the canopy.

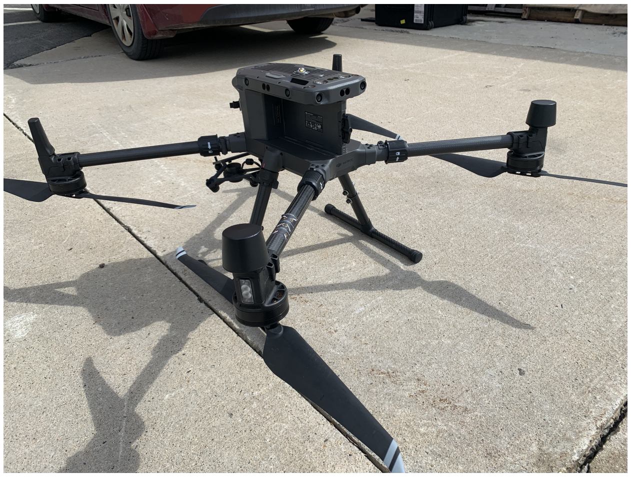


Figure 3.4: DJI Matrice 300

3.3 LiDAR Post Processing

After the point cloud data is obtained, further processing is required to produce 2D altitude surface maps. The overhead flight passes produce two files: (1) a LiDAR data file (.LAZ) which contains the LiDAR point cloud data and (2) a file containing text descriptions of the position created by GPS devices (.POS) which are integrated into the LiDAR measurements. These files are used to integrate the point cloud data and correlate it to the UAV's trajectory to build a complete 3D model of the forest. The data is corrected for the orientation of the UAV to take into account the LiDAR's mounting on the frame of the platform. This corrects for when the UAV pitch angle (to move forward) causes a discrepancy in the LiDAR point cloud because it is no longer oriented perpendicular to the ground. Once these corrections are applied, the full 3D data set is produced.

The creation of the 3D Digital Elevation Models (DEM) is completed using LiGeoreference software and Cloud Compare software. The data is first reviewed for accuracy, where data points with potential of high errors are removed. These "bad" data points are usually located at the corners of flight lines where there is an abrupt change in UAV direction and in points where the angle of incidence of the LiDAR beam is greater than 25°. High incidence angles result in greater beam divergence. The incident beam has an oblique shadow on the ground which causes loss in signal strength when the reflection is received. Any data point which exceeds five standard deviations in elevation compared to its ten neighboring points are removed. The consideration for the physical offset between the IMU and the LiDAR, as well as the systematic offsets in roll and pitch between LiDAR swaths, are taken into account using a bore-sighting routine in the LiGeoreference software. This step reduces errors in the data points collected during the straightaway of the flight path. The final point cloud is gridded to various spatial resolutions by averaging the returns that fall within each grid cell. The last step before creating the DEM is to identify the ground return from the LiDAR point clouds. The point cloud is normalized according to LiDAR ground returns. The elevation difference from those points to all other returns directly above them is calculated. This 3D point cloud results in a model where the LiDAR returns are colorized according to altitudes above the ground return points.

Point clouds are generated for Kingman Farm, Thompson Farm, and Ox Bow Farm. Using the GreenValley LiDAR 360 software, the point cloud of each test site is analyzed to extract the forest metrics from the point clouds. More information about the LiDAR software is available from the manufacturer [33]. The software identifies individual trees and calculates tree metrics for each site. The tree density, average tree height, average crown area, and average crown volume is calculated. These metrics are indicative of the level of obstacle density in the sub-canopy regions. An environment with excessive density will degrade GPS signals and UAV navigation will not be possible. The forest metrics for each site are shown in Table 3.1.

Site	Tree Density (trees / acre)	Average Tree Height (m)	Average Crown Area (m^2)	Average Crown Volume (m^3)
Kingman Farm	80	20.8	49.5	550
Thompson Farm	110	24.3	49.4	638
Ox Bow Farm	125	18.2	49.3	573

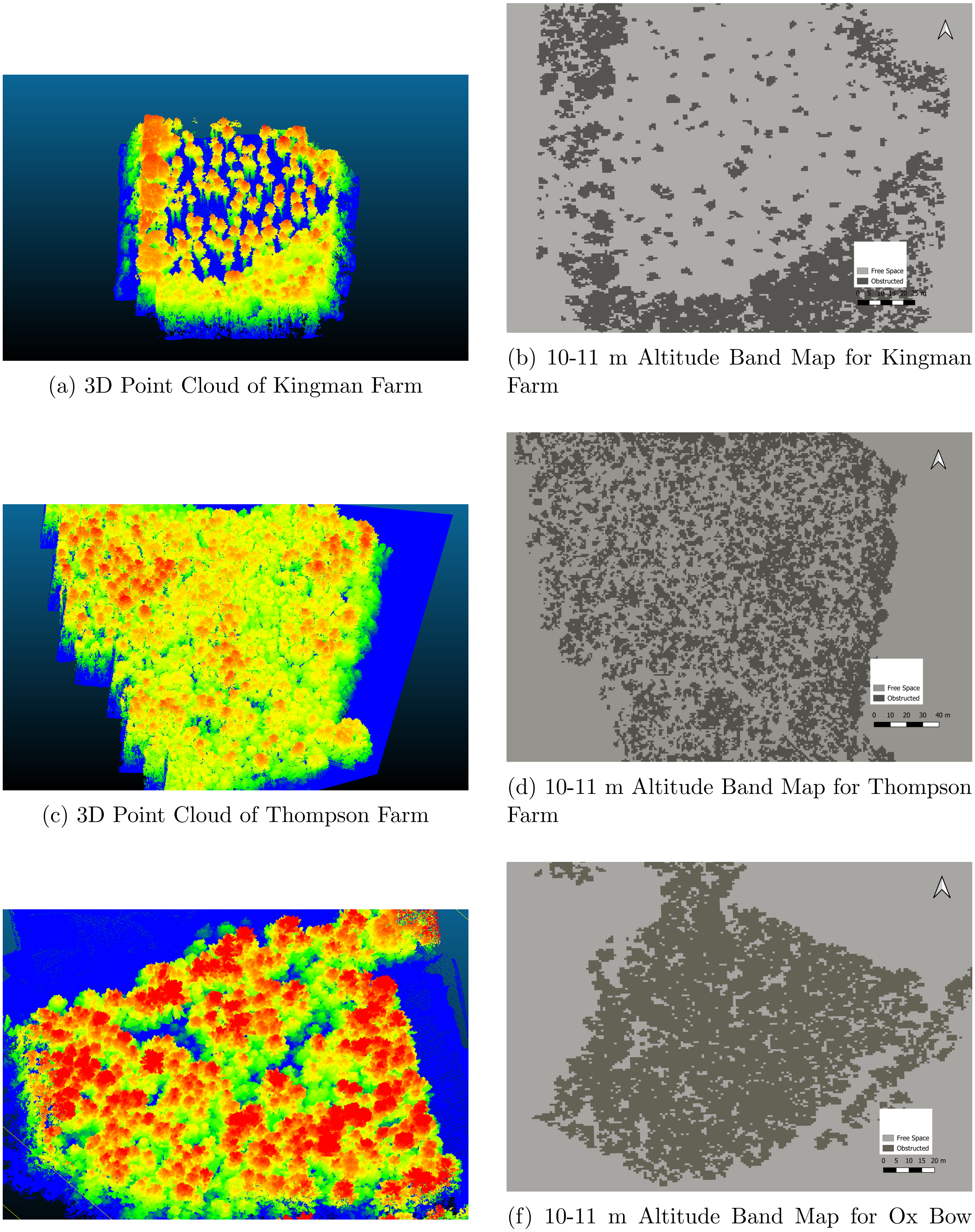
Table 3.1: Tree Metrics from LiDAR 360 by Test Sites

The average tree height, average crown area and average crown volume are very similar for each site. The tree density differs among the sites. Kingman Farm is the least dense, followed by Thompson Farm and then Ox Bow Farm. In the next section, the forest canopy profile is analyzed in order to find a test site that has adequate unobstructed space that allows for safe UAV navigation for below-canopy flight.

3.3.1 Generation of Obstacle Maps for A* Implementation

The DEM is used as a reference surface to create the 2D altitude specific surface maps which describe the obstacle orientations at specific heights above the forest floor. The points in the DEM are classified according to altitude as recorded by the LiDAR sensor. A simple algorithm is used to calculate the difference in elevation between the ground return and points that lie directly above the ground. The altitude is categorized from the forest floor to the top of the canopy. From the aforementioned discretization, all 1x1 m pixels within the same 1 m altitude band are grouped together to form a map by altitude beneath the canopy of the forest. The 2D surface maps (saved as .tif files) are viewed using python scripts referencing the DEM package (for Linux systems). These .tif files contain the intensity of the LiDAR return at each pixel, which indicates the strength of the reflected signal as received by the Livox sensor. Some returns are stronger than others. The return can be color-mapped for visualization purposes. The pixels where the intensity is 0 are the locations where no obstacles exist. Any pixel with an intensity greater than 0 is assumed to be obstructed by an object in the forest.

This process is repeated for all three test sites. The elevation band from 0 to 1 m and the 1 m band immediately below the canopy crown are very densely packed at all three sites. Navigation in these elevation bands is not be possible. There is, however, open airspace in the elevation bands that lie between the canopy crown and the forest floor specifically in the 10-11 m altitude bands along with the 3D point cloud for the three test sites is shown in Figure 3.3. It is clear that open connected space is available for both Kingman Farm and Thompson Farm. Ox Bow Farm has very little free space and almost no connected free space.



(a) 3D Point Cloud of Kingman Farm

(b) 10-11 m Altitude Band Map for Kingman Farm

(c) 3D Point Cloud of Thompson Farm

(d) 10-11 m Altitude Band Map for Thompson Farm

(e) Image 5

(f) 10-11 m Altitude Band Map for Ox Bow Farm

Figure 3.5: 3D Point Cloud and Subsequent Altitude Band Map For Three Test Sites

The goal is to have A* use these altitude layers to determine a clear, sub-canopy area in which the UAVs can safely maneuver for remote sensing of snow depth. The obstacle percentage, which calculates the ratio of free pixels to the total number of pixels for a given altitude band map in a forest is presented for the three test sites.

$$C_r = (\frac{OP}{TA}) * 100\% \tag{3.1}$$

where OP and TA are the number of obstructed pixels and total pixels, respectively, at any given forest altitude band map. The minimum value of the obstacle percentage is found, and the corresponding altitude is considered to be the safest UAV flight altitude.

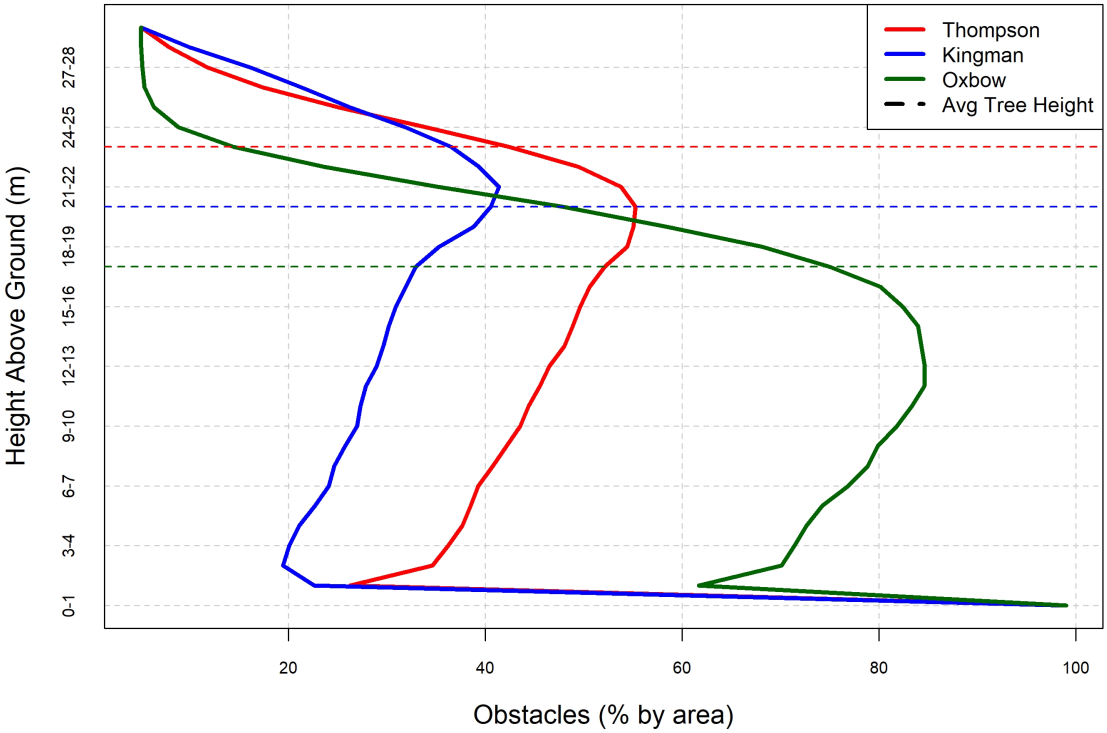
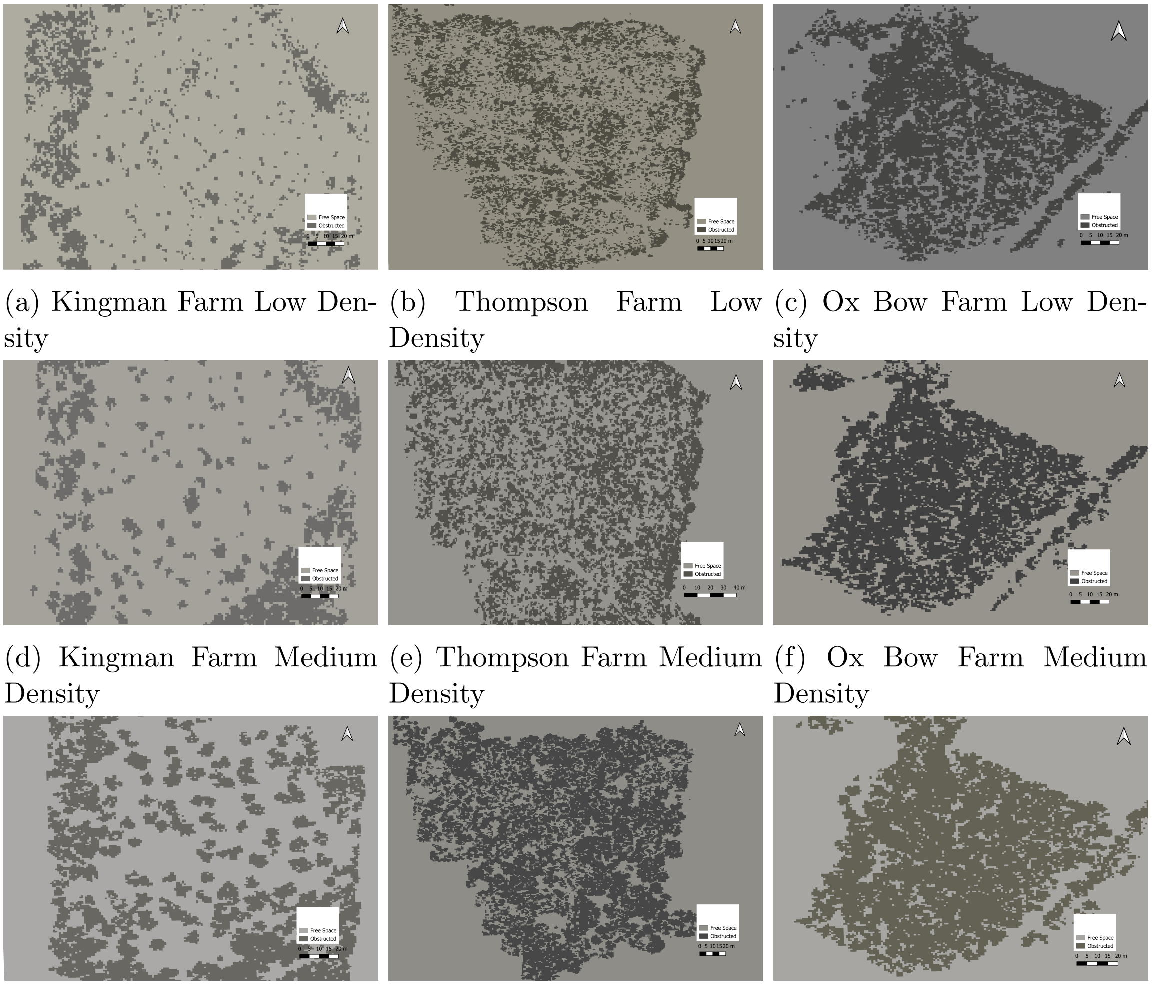


Figure 3.6: Obstacle Percentage by 1m Altitude Bands and Average Tree Heights for Three Test Sites

Figure 3.6 shows that all three sites exhibit similar obstacle percentage patterns as the height increases. The forests are densest at the base of the forest. The number of obstacles declines as the altitude increases until a minimum for the obstacle percentage is reached. For all three sites the least obstructed altitude band occurs between 2 and 4 m. Above 4 m, the ratio begins to increase in a relatively linear manner till the average crown height is reached, marked by the dashed lines showing the average tree heights for each site. The maximum occurs within 1 m of the average tree height for Thompson and Kingman Farm but for Ox Bow Farm it occurs 7 m below the average tree height. The obstacle percentage begin to decline above the average tree height but this airspace is not desirable for flying because the canopy crown would obstruct remote sensing devices. The Newmarket property is the densest of the three sites, which is supported by the tree density measured in Table 3.1, followed by Thompson Farm then Kingman Farm. UAVs that are placed at the ground at Ox Bow Farm and Thompson Farm connected to less than 10 satellites. The Kingman Farm property is the forest chosen to conduct proof of concept testing because it is the least densely packed site and an average of 14 satellite connections are present at the forest floor. The sparse canopy will insure minimal degradation of GPS signal. More accurate GPS denied localization methods are needed to enable UAV navigation in the other two sites.



(a) Kingman Farm Low Density

(b) Thompson Farm Low Density

(c) Ox Bow Farm Low Density

(d) Kingman Farm Medium Density

(e) Thompson Farm Medium Density

(f) Ox Bow Farm Medium Density

(g) Kingman Farm High Density

(h) Thompson Farm High Density

(i) Ox Bow Farm High Density

Figure 3.7: Altitude Map Comparison for Three Sites

Figure 3.7 shows the altitude maps of low, medium and high density altitude bands for each test site. In Figure 3.7 the rows show the altitude band maps for each site for the same density and the columns show the altitude band maps in each test site in order of increasing density. The columns from left to right show Kingman Farm, Thompson Farm, and Ox Bow Farm respectively. The low density altitude band maps are representative of the 2-3 m altitude band maps for all sites. The medium density maps for Kingman Farm, Thompson Farm and Ox Bow Farm, represent the 12-13 m, 10-11 m, and 6-7 m altitude band maps respectively. The high density map for Kingman Farm, Thompson Farm, and Ox Bow Farm represents the 21-22 m, 20-21 m, and 10-11 m altitude band maps respectively. Kingman Farm exhibits the most free space across all levels of densities, while Ox Bow farm has very little free space at all density levels. For these reasons the 2-3 m altitude band map (Figure 3.8) is selected to demonstrate sub-canopy navigation of UAVs.

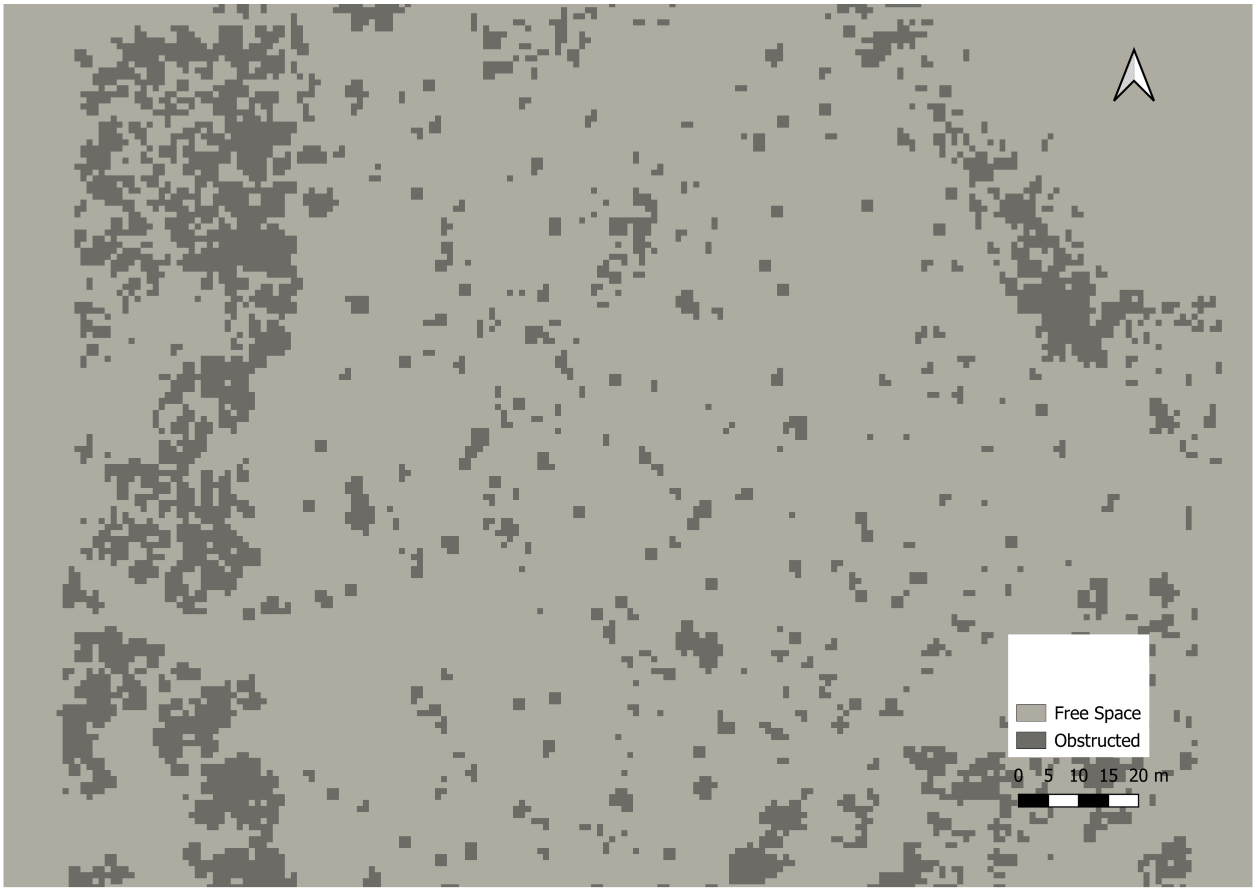


Figure 3.8: Kingman Farm 2-3 m Altitude Band Map Showing Open Space and Obstructed Space

CHAPTER 4
UAV Test Platform

4.1 UAV Experimental Equipment

4.1.1 UAV Platform

The test platform for multi-UAV flights used in this study is the Holybro S500 V2 UAV platform (shown in Figure 4.1). This platform is compatible with a Pixhawk flight controller (Figure 4.2) and can be used for autonomous survey missions. The platform dimensions are provided in Table 4.1.

Dimensions	383 X 385 X 240 mm
Wheelbase	480 mm
Weight	985 grams

Table 4.1: Holybro S500 V2 Specifications

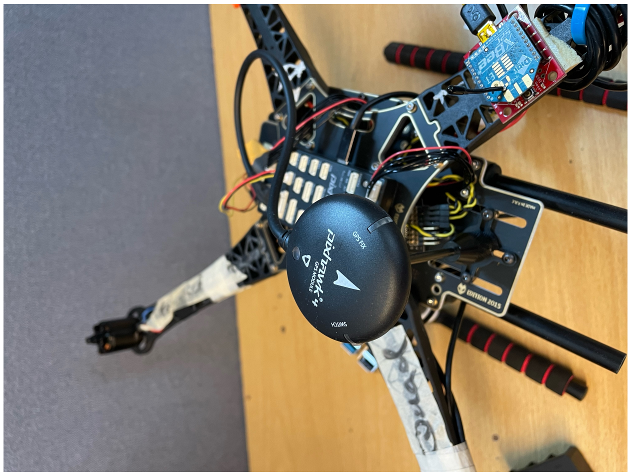


Figure 4.1: Holybro S500 V2 UAV

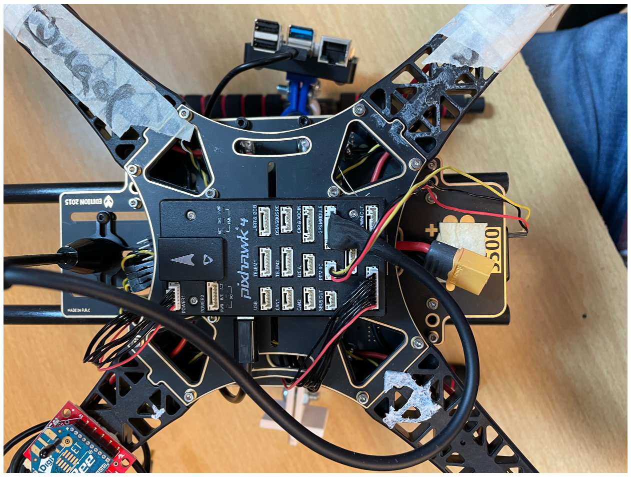


Figure 4.2: Holybro S500 V2 UAV

This S500 platform is chosen because of its cost efficiency (approximately $600) and the open-source Pixhawk firmware used to control it. The system comes with all required hardware, along with relatively straight forward instructions for assembling and operating the platform. Most of the system is held together by 2 *mm* Allen bolts making for convenient replacement when necessary. The components of the S500 kit are provided in Table 3.3.

Pixhawk 4 autopilot with IMU inside and Pixhawk 4 GPS
Power management board
Four rotor arms made of high strength plastics
4 motors
BLHeli S electronic speed controller
4 plastic propellers
Telemetry radios

Table 4.2: Contents of Holybro S500 V2 Development Kit

4.1.2 UAV Control Components

The Pixhawk 4 autopilot system is an open-source flight controller developed for autonomous UAV control. It is compact (44 x 84 x 12 *mm*) and weighs 15.8 *g*. The Pixhawk has functionality that allows for off-board control as well as manual Radio Control (RC) for Vertical Takeoff and Landing (VTOL) UAV systems. The main flight management unit consists of a 32-bit ARM Cortex M7 processor with a 216MHz clock speed, 2 MB of memory and 512 KB of RAM. The input and output signals are managed by an ARM Cortex -M3 processor. The Pixhawk has an accelerometer, gyroscope, magnetometer, and barometer integrated into the circuitry for orientation and position measurements. These sensors are initially calibrated using the QGroundControl software. This application is a free open-source software that communicates with the Pixhawk flight board using a 915 MHz telemetry radio. The Pixhawk 4 is preinstalled with PX4 firmware which uses the MavLink communication protocol to send calibration parameters to and from the UAV and a QGroundControl installed base computer. The Pixhawk has an 8-pin Pulse Width Modulated (PWM) output for the motor controllers, an output for radio control based uses, and a GPS input port for navigation. The GPS that comes standard with the kit is a U-blox M8N, and it has accuracy of approximately ±2.5 m. Real Time Kinetic positioning (RTK) GPS is also supported by the Pixhawk and is recommended for high-precision surveying missions.

4.1.3 UAV Software Modules

The Holybro S500 system is equipped with a Raspberry Pi 4 microprocessor (shown in Figure 4.3) with 8 GB of RAM to run off-board commands and communicate with the Pixhawk autopilot for navigation. The Ubuntu 20.04 operating system is installed on the microprocessor along with the full Robot Operating System (ROS) communication protocol to enable communication with the Pixhawk 4 system.

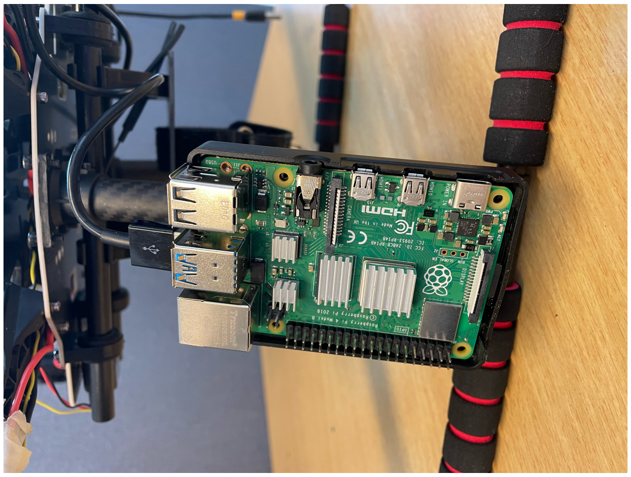


Figure 4.3: Rasberry Pi integrated with Holybro S500

The ROS network allows the user to access all the sensor information that the Pixhawk receives and to send necessary control inputs to manipulate the UAV platform during missions. The diagram in Figure 4.4 shows the ROS communication network architecture.

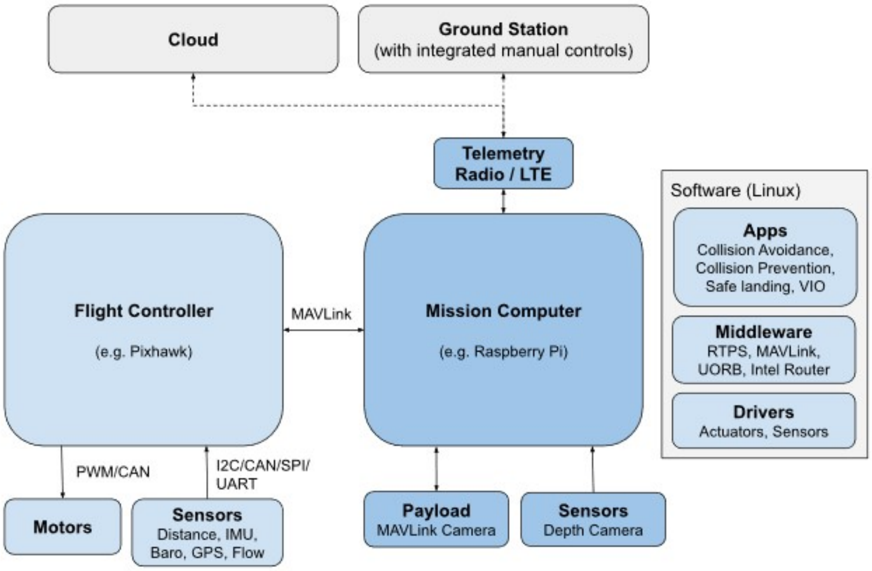


Figure 4.4: ROS and PX4 Communication Architecture [3]

Ubuntu is installed on the Raspberry Pi along with ROS and MAVROS. MAVROS is a ROS package that allows the MavLink protocol, native to the PX4 firmware, to communicate with ROS based computers. The next step is to physically connect the Raspberry Pi to the Pixhawk 4 autopilot using a micro-USB port on the Pixhawk 4 board. ROS communication with the PX4 firmware is initiated on the mission computer by running a launch file developed for this communication with the syntax of "roslaunch mavros px4.launch". This command initiates an instance of ROS and MAVROS to automatically start reading sensor messages from the Pixhawk. This is the basis from which all off-board flight control operates. Python packages for ROS communication allows for the development of Python scripts that can use these ROS messages to manipulate the flight behavior to specific user needs. It should be noted here that the Pixhawk is notably user friendly but does not allow for user flexibility. For example, the waypoints may be easily programmed, but the on-board UAV control algorithm is not accessible and at this point in time, cannot be easily modified. For more information on quadrotor dynamic modeling and control, the reader is referred to Appendix A.

4.2 UAV Software

4.2.1 A* Path Planning Code

For this research, the A* path planning algorithm is implemented using Python for reasons of convenient implementation and integration into the flight management systems. The first object that needs to be defined for the implementation is a "node" class which allows the computer to recognize each discrete pixel and store information about the pixel. This information includes its f, g, and h score, its classification as an obstacle space or free space, its position in the search space map, and the position of its adjacent pixels (so that the heuristic scores can be compared during the path search). The start node is a user-defined input. Its g score is inherently initialized at 0 because this value is a measure of its distance from the start node. Its h score is calculated for the start node based on, for this research, the Manhattan distance from the end node, which is also a user-defined input. There are two lists created for each node: (1) an open list which is populated with the neighboring nodes that have not been compared against the heuristic (defined in Chapter 3) and (2) a closed list which is populated with the adjacent nodes that have been compared. Each f score in the open list is compared to the current node's f score. If the value is less than that of the current node, then that corresponding node is removed from the open list and added to the closed list. When all the neighbors have been checked, a priority queue is used to sort the closed list. Here the lowest f score in the closed is identified, and the corresponding node becomes the current node to be compared. If the current node is the "end" node, then the path is considered found and all the nodes prior to the "end" node are recorded as the waypoints of the path to be followed from the "start" node to the "end" node. The pseudo code for A* planning algorithm is displayed below..

```
If f.neighbour_node < f.current_node
Append neighbour.node to closed list
```

```
Lowest f score in closed list = current node
Add current node to path vector
If current node == end node
        Path has been found
        Else keep checking
END
```

The “start” and “end” nodes are checked to ensure they are open spaces (i.e, not obstructed). Because of the h score, only nodes that are likely to lead to the goal are checked. This approach makes the path finding computation time quicker than brute force search algorithms.

4.2.2 UAV Flight Control Code

The flight control code is implemented in Python to make use of an in-built Python package (rospy) that can directly access and publish to ROS topics. The code for flight control is an iterative loop that reads a data structure containing the waypoints produced by the A* path planning method and executes each successive waypoint until the mission is complete. The ROS network provides subscriber nodes to read sensor messages regarding the UAV current state and publisher nodes to send control inputs to the Pixhawk board to appropriately alter the current state to the next desired state.

The mavros subscriber messages that are of importance for the flight code include “mavros/state”, “mavros/global_position/global”, and “mavros/altitude”. The first message monitors the PX4 state variable which allows the user to switch between flight and ground modes. The second message is the incoming reading from the GPS module which indicates the current location of the UAV. The third message monitors the altitude (above sea level) according to the barometer in the Pixhawk 4 autopilot system. The publisher message that is used during waypoint control is “mavros/setpoint_raw/global”. This message allows the flight computer to command the UAV to fly to the desired setpoints and defines the desired latitude, longitude, and altitude of each waypoint.

On initialization of the control code, the system checks the current position of the quadcopter and continues measuring the current location until the standard deviation communicated in the incoming GPS messages drops below a user-defined threshold, allowing the GPS sensor time to settle to its steady-state reading and to ensure that incorrect coordinates are not being used. Once the current position is estimated, the flight code begins a counter and compares the difference between the current and desired position. The PX4 autopilot is then instructed to send the UAV to the desired waypoint if the GPS error threshold is not reached. Once the threshold is reached, the counter is increased by 1 and the next waypoint is used as the setpoint. Once the last waypoint is reached, the mission is considered as successfully completed, and the UAV is navigated safely "home" (e.g. above-canopy).

4.2.3 Integration with A*

The forest altitude-specific 2D maps are uploaded to the mission computer for path planning. A* is implemented and the appropriate Python code is uploaded to the on-board flight computer. On initialization of the system computer, an instance of ROS is started, and Pixhawk sensor messages are checked to ensure the system is correctly reporting information. The A* code is remotely executed to produce the navigation path.

The A* implementation requires the 2D map of the altitude space and the local grid cell coordinates of the mission start and end points. The code output is a list of the latitude-longitude pairs of each pixel location along the path produced within the map. This list is used for navigation during the mission. These latitude-longitude pairs are in the World Geodetic System 1984 (WGS84) reference datum.

During the autonomous mission, a separate flight control script is used to monitor UAV flight sensor feedback from the Pixhawk and to compare them to the desired locations of the path. Here, the control script utilizes the ROS network's subscriber and publisher scheme to read sensor messages and then send appropriate flight control input to effect desired behavior. For example, subscriber nodes in the ROS network can access PX4 firmware information to monitor UAV position, velocity, orientation, and altitude.

Allowed tolerances for the tracking error are inputted a prori in the flight control script to ensure mission success, such as these shown in Table 4.3. These thresholds are adjusted according to factors such as pixel size in the search map, GPS resolution, and dimensions of the UAV platform.

Threshold radius for reaching waypoint vertically	0.25 m
Threshold radius for reaching waypoint horizontally	0.25 m

Table 4.3: Flight Code Thresholds

CHAPTER 5

Preliminary Testing

5.1 Initial Numerical Simulations

The flight code is tested using the Gazebo simulation software. This software enables users to test flight code quickly and check for errors. Gazebo runs the PX4 firmware virtually and enables the ROS network to communicate with this virtual instance. It also publishes the same topics which the flight code references (Chapter 4.2.2). Gazebo comes integrated with the Open Dynamics Engine (ODE) which allows for accurate simulation of the UAV system dynamics with real-world physics.

The Gazebo simulation software allows users to simulate the hardware, software and the physics of the environment in which UAV systems are tested. Gazebo helps to save time by enabling users to test all aspects of the UAV system before actual field testing. A test path for the UAV is produced using A*'s path finding algorithm on the 2- 10 m altitude band map. The simulated flight speed is set to 2 m/s and the altitude is set to 8 m. The generated path is relayed to the simulation software. The mission completion time is simulated at a 1:1 scale. Figure 5.1 shows the planned flight path and the actual flight path.

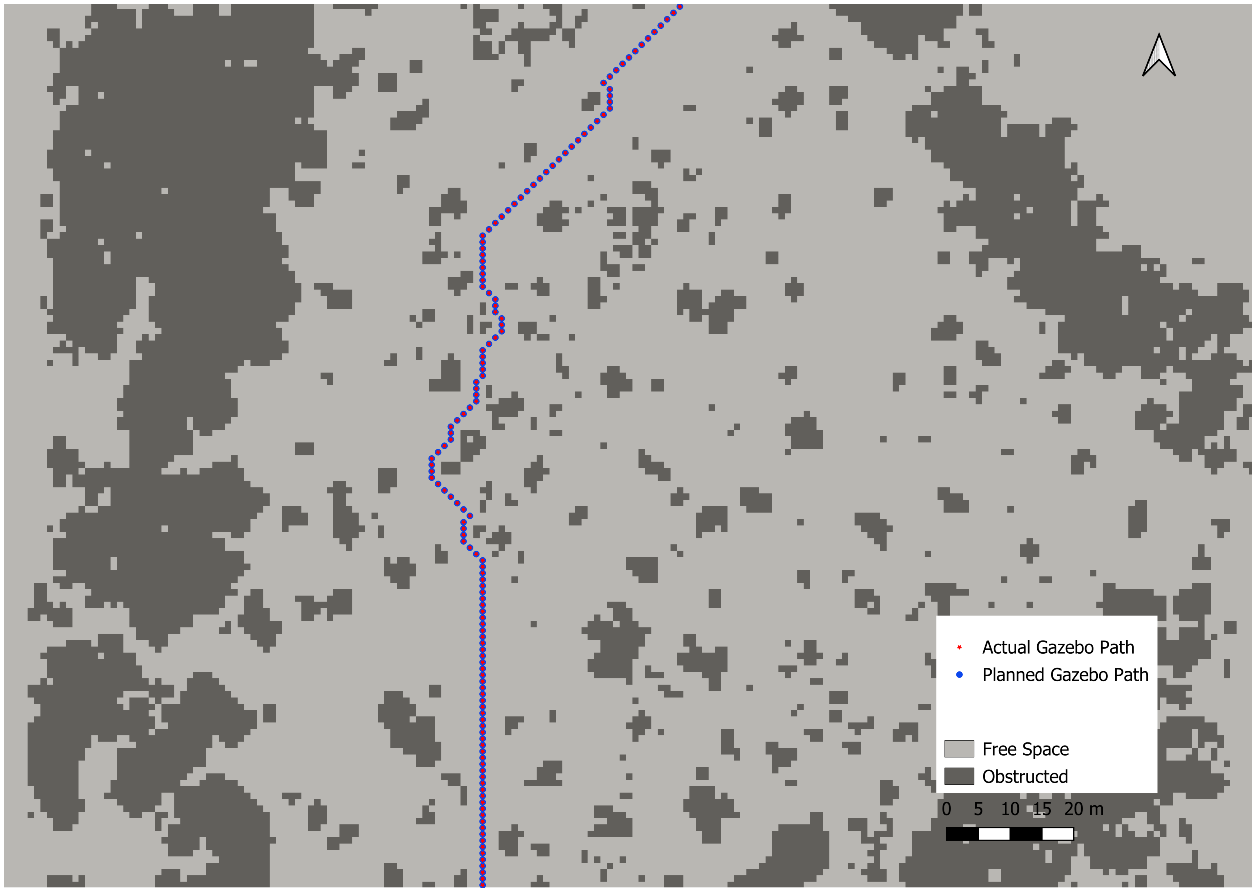


Figure 5.1: Gazebo Simulation Planned and Actual Path for Kingman Farm

These simulation tests show that the flight control code successfully commands the UAV to travel through successive waypoints. The flight dynamics are also shown to be stable. The ROS and PX4 communication systems are able to manage message streams with its available on-board RAM. Gazebo, however, does not simulate sensor errors and assumes all feedback to be perfect. Therefore, the planned path and actual path (Figure 5.1) are numerically identical to each other. This will not be the case for an actual UAV flight because there are many sources of sensor error. The next step is to test the UAVs in real-world environments to understand sensor flaws.

5.2 Experimental Testing Under Controlled ”Laboratory” Conditions

Preliminary field tests are conducted in the empty UNH Olsen Advanced Manufacturing Center parking lot located in Durham, New Hampshire (Figure 5.2). This laboratory site is chosen to allow for safe experimental testing in an artificially simulated forest environment, specifically to check hardware functionality and controller performance.

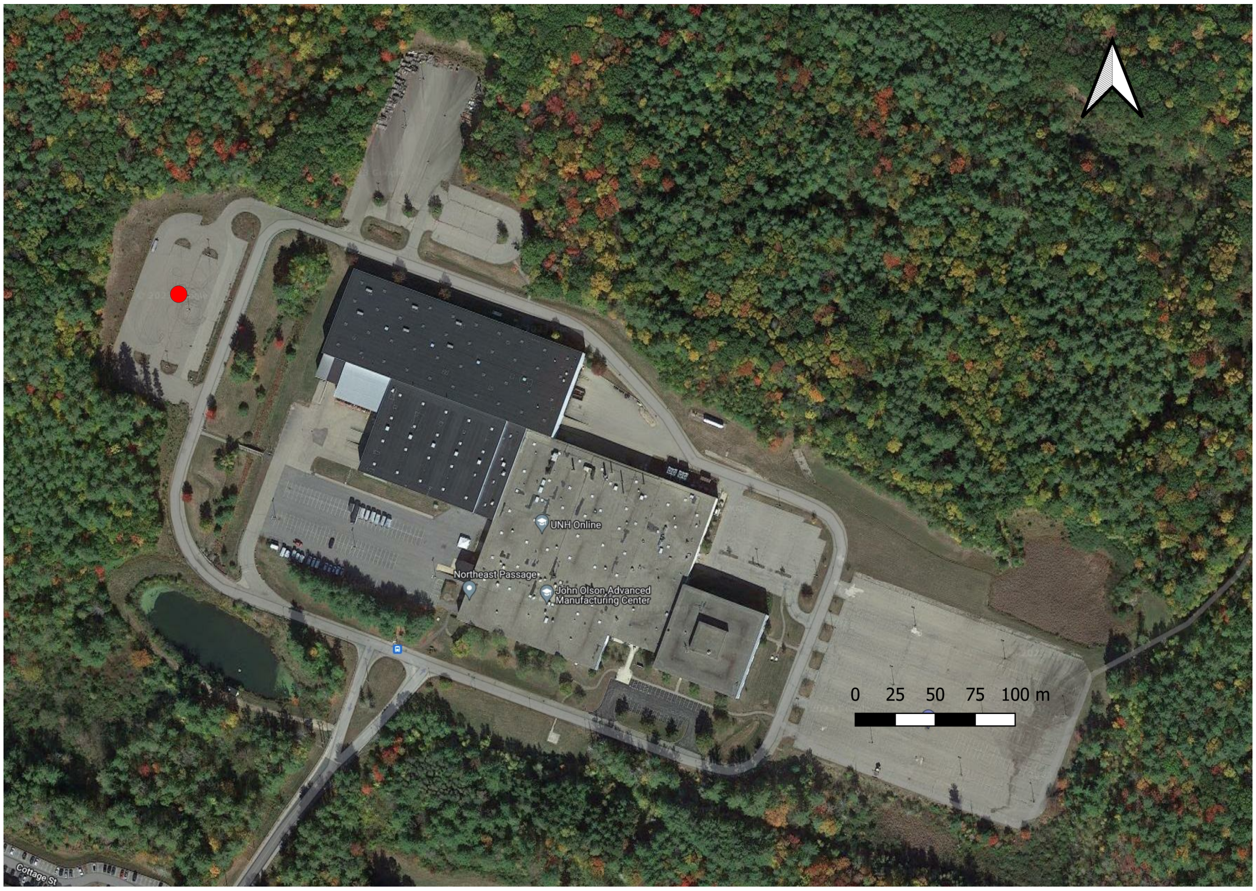


Figure 5.2: UNH Olsen Advanced Manufacturing Center Test Site (Red Dot Showing Parking Lot)

2D altitude maps of Kingman Farm are virtually overlaid over the geolocation of the parking lot. In this way, the obstacles of the forest are artificially injected while maintaining a clear and safe environment. A key goal is to observe the UAV GPS error associated with the Holybro S500 platform. The real time tracking error is evaluated using a Xsens mti g710

GNSS system, which is considered as ground truth. Both sensors are shown in Figure 5.3. The Xsens mti g710 is rated to have sub-meter accuracy. The navigational Pixhawk GPS is rated to have 5 m accuracy. The two GPS measurements are recorded during multiple flights to judge the accuracy of the Pixhawk 4 GPS system. The flight data for one example test is shown in Figure 5.4. A total of seven different flight tests are conducted. The results can be found in Appendix C.

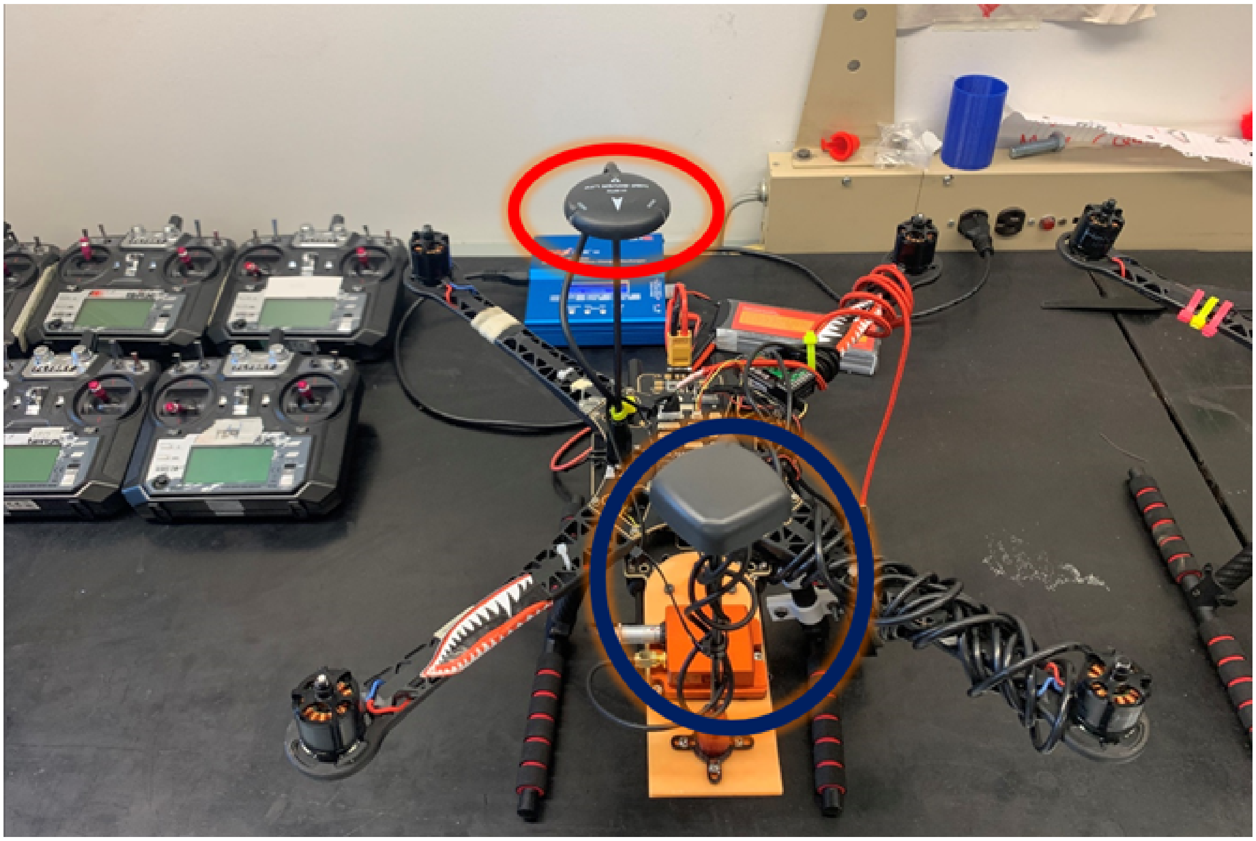


Figure 5.3: Xsens Mti g710 GPS (Blue Circle) and Pixhawk 4 GPS (Red Circle)

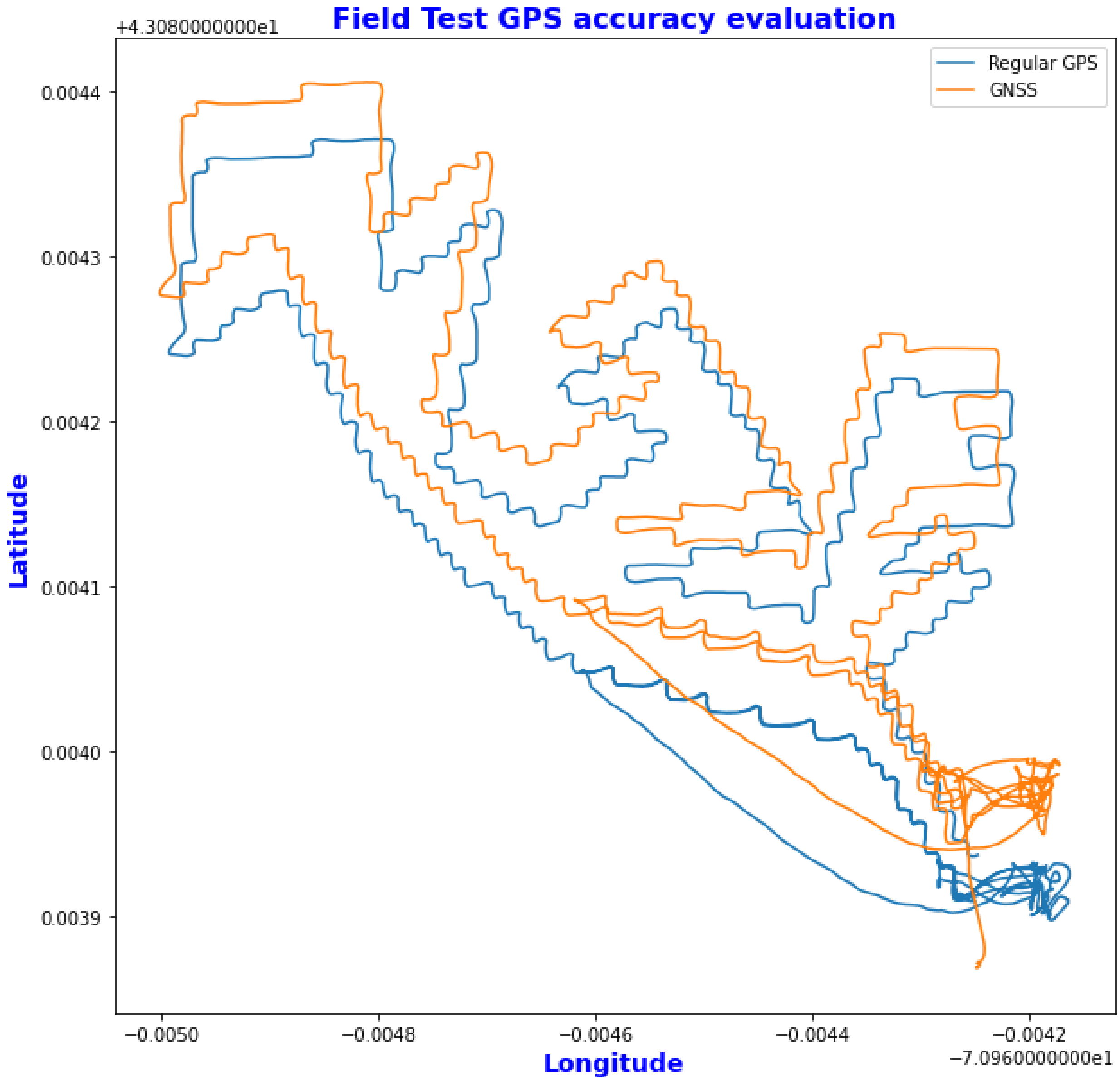


Figure 5.4: Test 1 at UNH Olsen Advanced Manufacturing Center Parking Lot Comparing Xsens (Orange) and Navigational GPS Measurements (Blue)

The PX4 GPS has a maximum measurement rate of 10 Hertz, while the Xsens GPS has a maximum measurement rate of 400 Hertz. Therefore, for the same flight the Xsens has more data points compared to that of the Pixhawk GPS. To calculate the difference between the two, the Pixhawk GPS measurements are linearly interpolated to extract GPS data that is compatible and directly comparable to that of the Xsens GPS data. The mean difference between the Xsens and PX4 GPS measurements is 1.38 m and the root mean square error (RMSE) is 1.46 m.

During the test flights in the parking lots, it is observed that the UAV system lowers its altitude by 1 to 2 m during wind gusts. This is presumed to be caused by the barometer reporting inaccurate altitudes following the change in pressure caused by the wind flow. The PX4 control system tries to compensate for the false altitude measurement and causes the UAV to change its altitude.

A 1 m altitude band cannot guarantee safe navigation because wind gusts can cause the UAV to drop in altitude. This can cause obstacles to appear in the flight path which have not been accounted for during the A* path planning. This barometer issue is addressed by increasing the altitude band of the 2D maps. The obstacles present from 2 to 10 m are combined to create a single 2D map. This can cause obstacles to appear in the flight path which have not been accounted for during the A* path planning. Large altitude bands cause more restriction of the air space because obstacles from many altitudes are being combined into one map. Figure 5.5 shows the 2-3 m altitude band map which has the lowest obstacle percentage for Kingman Farm. All other 1 m altitude band maps (5-6 m, 7-8 m, and 9-10 m) have a higher obstacle percentage and are shown in Figures 5.6-5.8. By consolidating all the obstacles in the altitude between 2 and 10 m (Figure 5.9), the total amount of free space is reduced. Another approach is to use accurate terrain following sensors that allow for path planning in a more restricted altitude band. This will increase the the amount of free space in each altitude map.

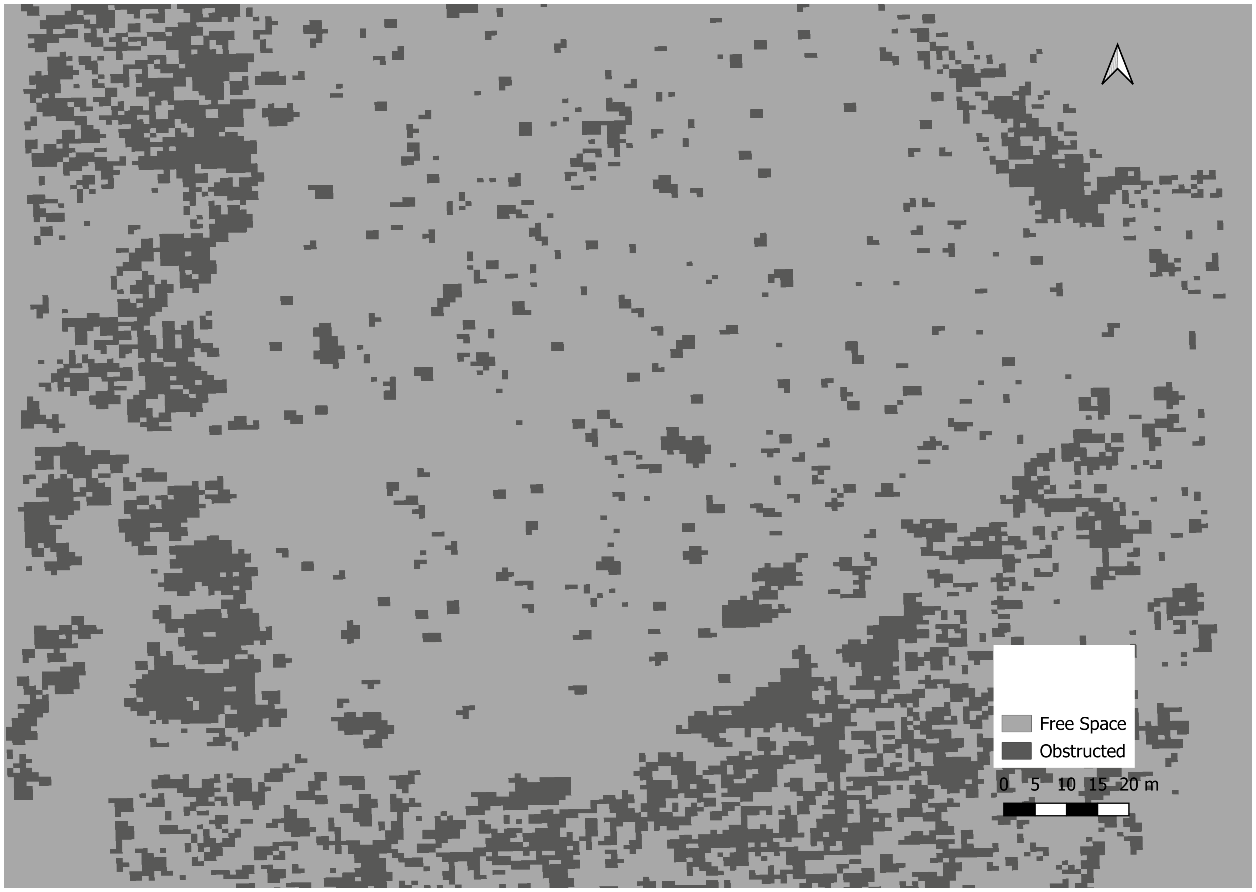


Figure 5.5: 2-3 m Altitude Band Map of Kingman Farm

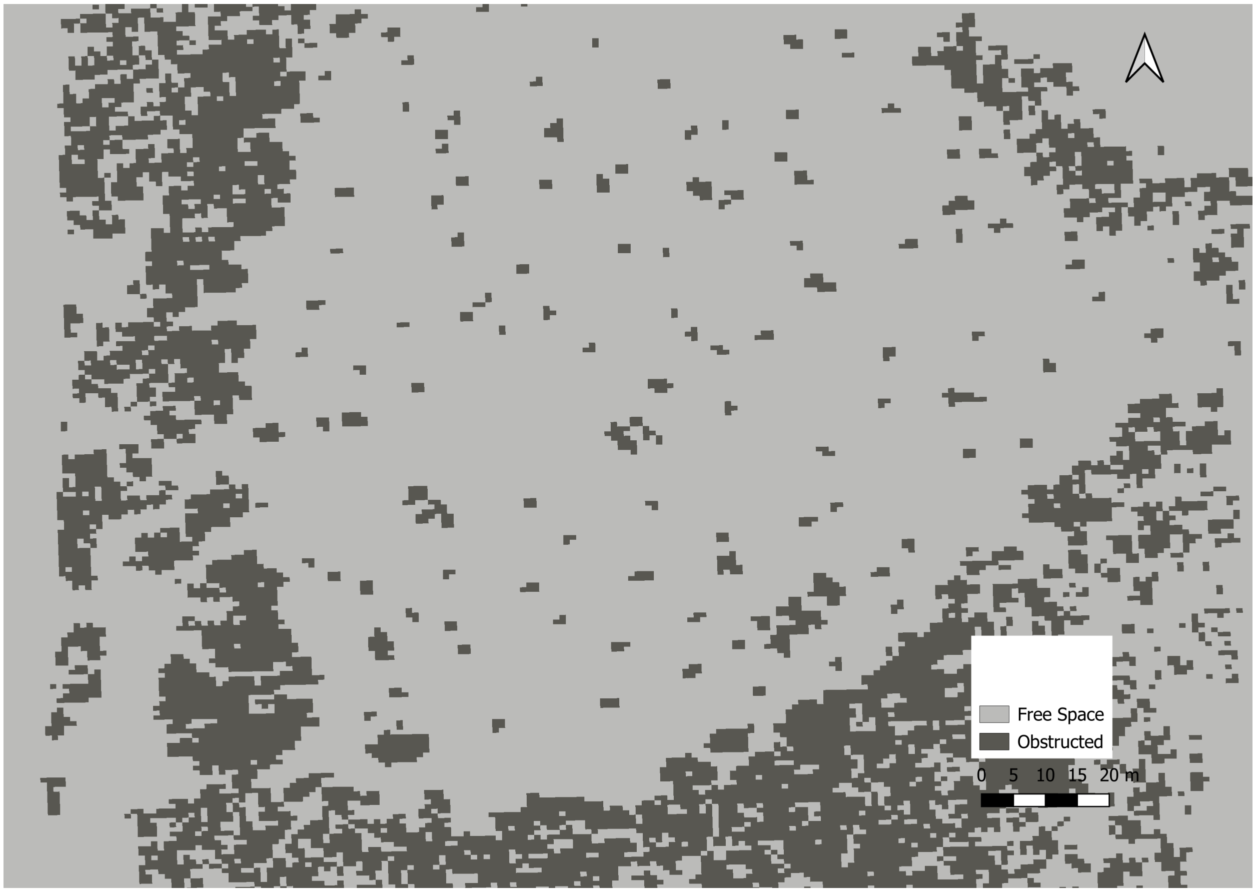


Figure 5.6: 5-6 m Altitude Band Map of Kingman Farm

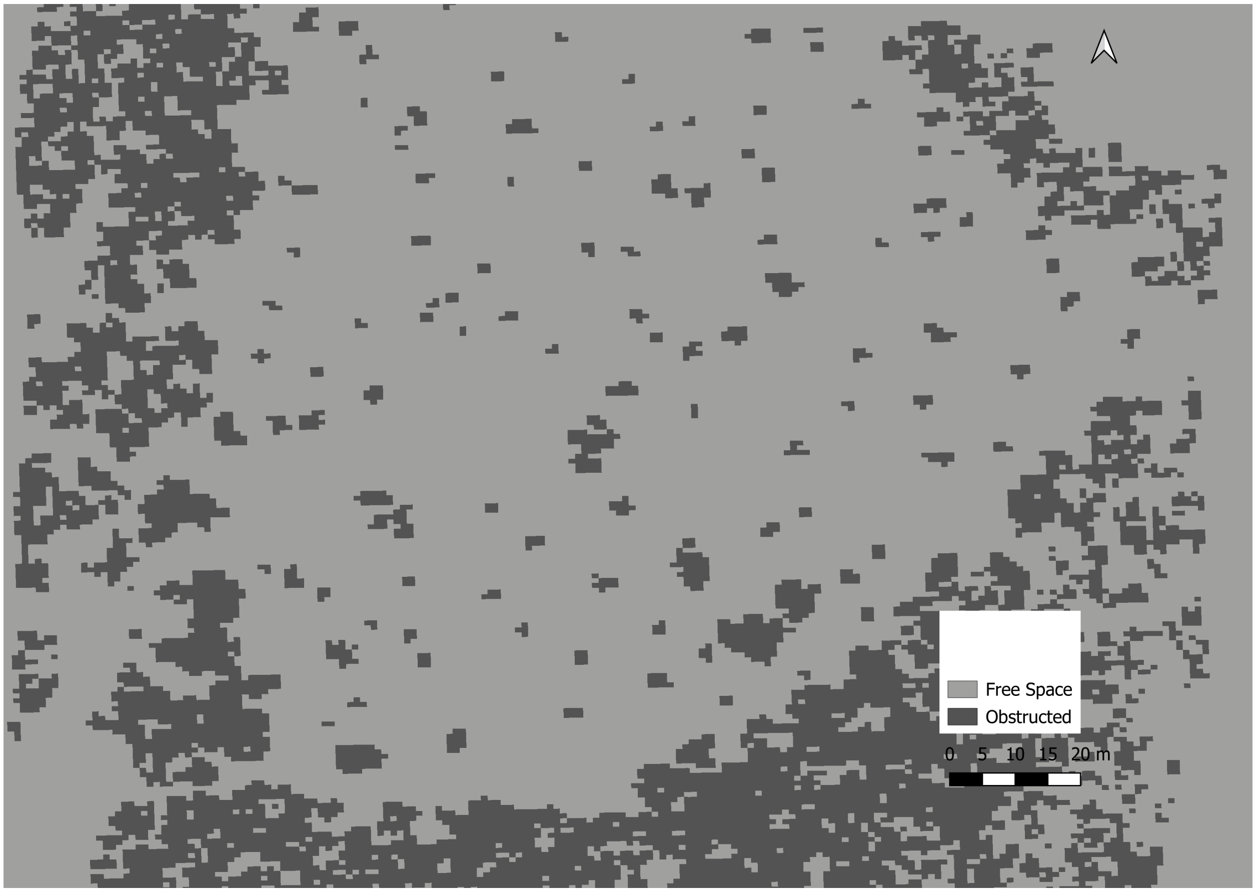


Figure 5.7: 7-8 m Altitude Band Map of Kingman Farm

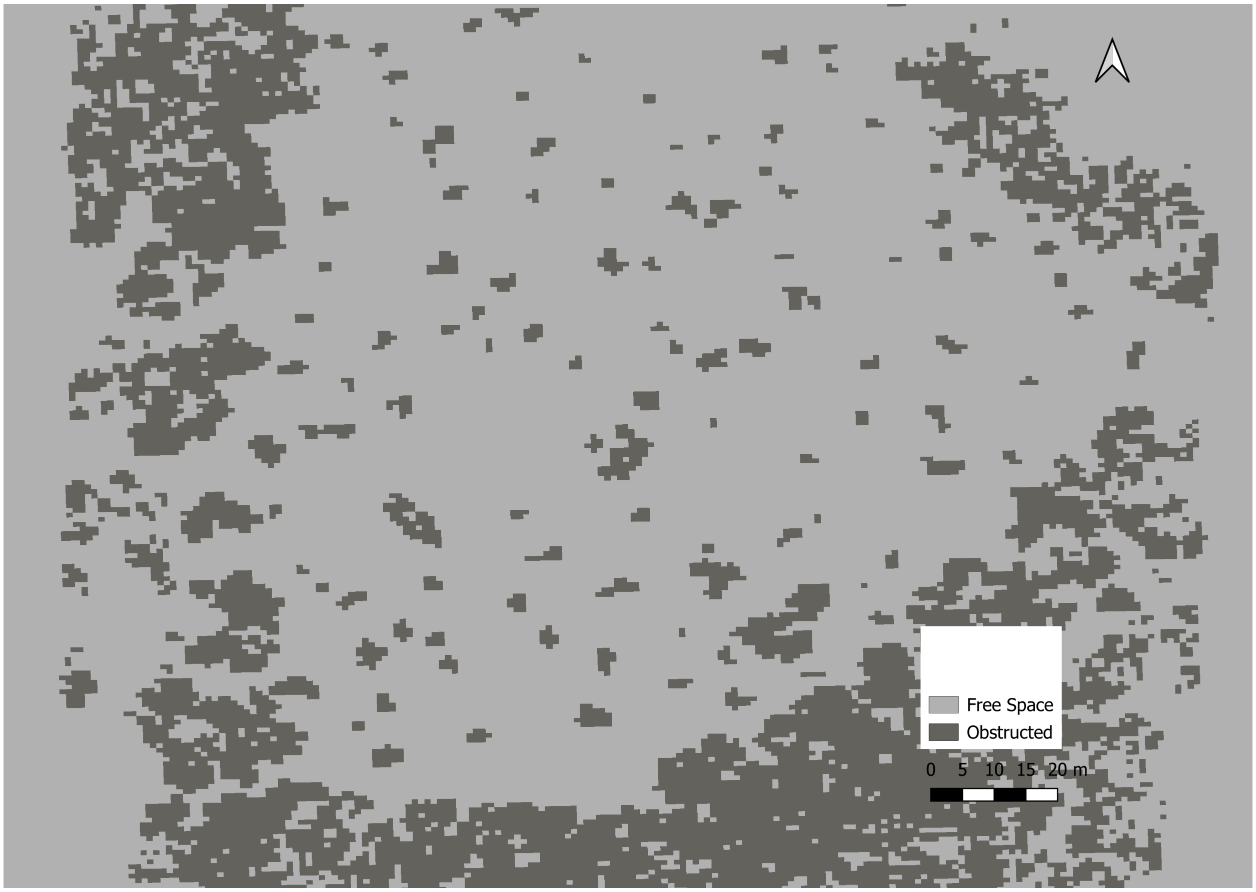


Figure 5.8: 9-10 m Altitude Band Map of Kingman Farm

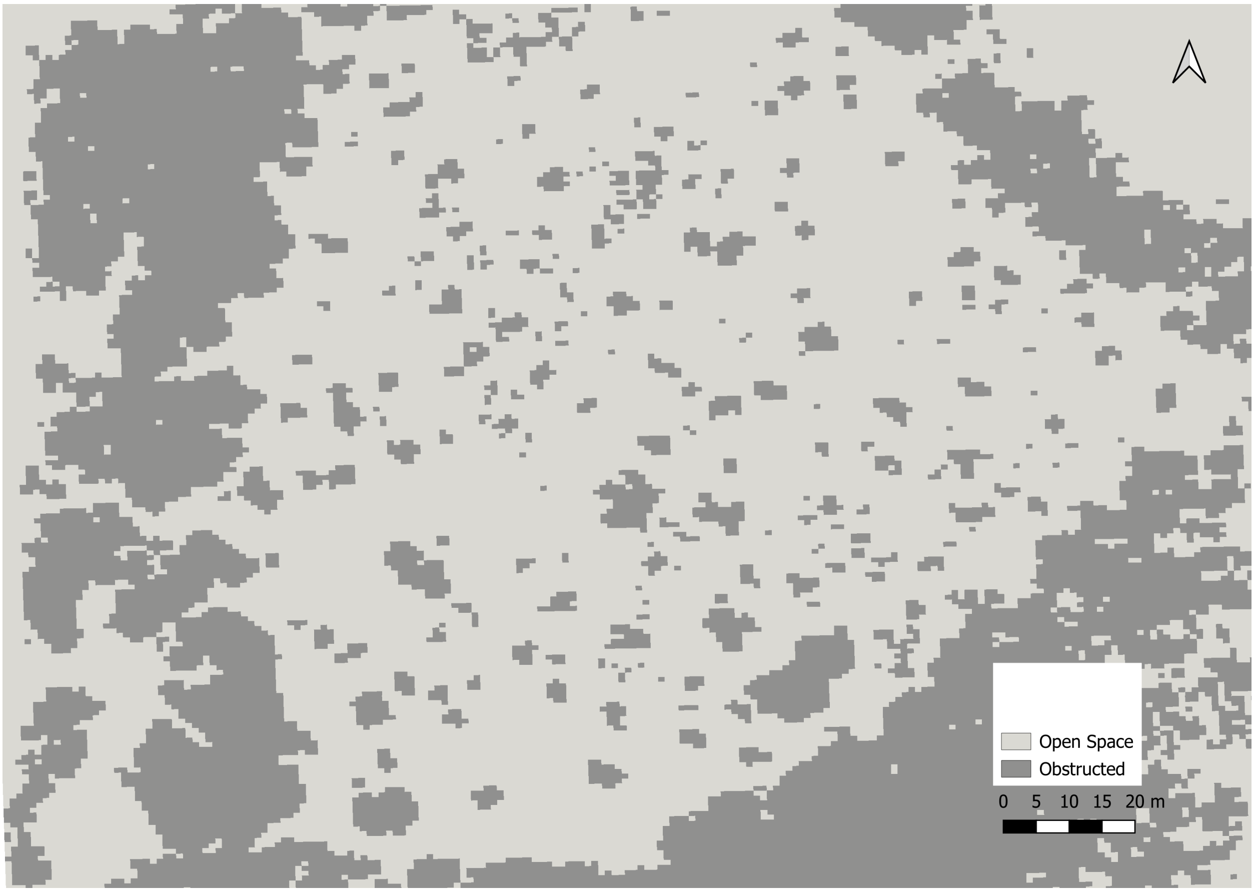
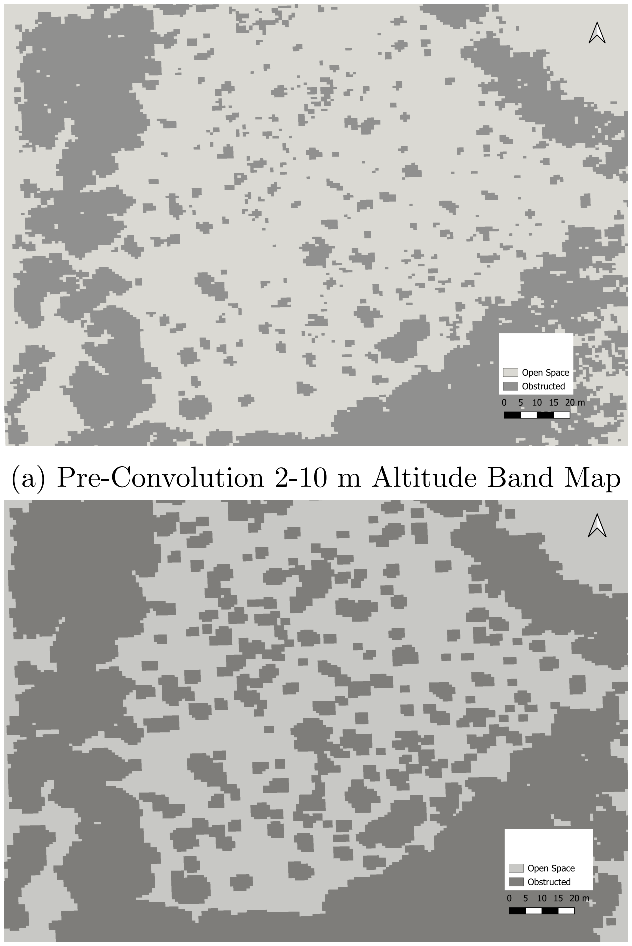


Figure 5.9: 2-10 m Altitude Band Map of Kingman Farm

From the parking lot test flight (Figure 5.4) it can be seen that the Pixhawk 4 GPS performs within its rated accuracy of 5 m, which is not sufficient for the forest environment. To overcome this problem, a virtual 2 m buffer is placed on the obstacles in the Kingman Farm forest 2D altitude maps. This buffer is created by performing 2-D image convolution on the altitude maps. Image convolution is a process in which elements of pixels are spread to their surrounding pixels using a weighted kernel. In this way, the obstacles are virtually inflated so that a safe buffer can be added to the maps. For this study, each pixel in the map that has an obstacle present is inflated by 2 m to compensate for GPS inaccuracy. This buffer is chosen because it does not constrict the entire airspace. Too large of a buffer can cause the 2D altitude maps to have inadequate free space. Field tests should be conducted in the forest to accurately decide on the appropriate buffer level. The two main factors when choosing the buffer size are the ability to compensate for GPS inaccuracies and ensuring enough free space to cover the search area. Figure 5.10 shows the search space before and after buffering obstacles. Figure 5.10(b) is the final map that is used to plan paths for UAV based snow depth data collection under the forest canopy.



(a) Pre-Convolution 2-10 m Altitude Band Map

(b) Post-Convolution 2-10 m Altitude Band Map

Figure 5.10: Pre-convolution and Post-Convolution 2-10 m Altitude Band Maps

CHAPTER 6

Field Testing of Autonomous Sub-Canopy Multi-UAV Flights

After software and hardware are tested using the methods described in Chapter 5, the UAV is deemed ready for testing at the UNH Kingman Farm test site described in Chapter 3. The tests are conducted from January to March of 2023 with snow present on the forest floor (Figure 6.1). Three Holybro S500 UAVs are used for this test. The main purpose of this test is to validate the path planning and navigational capabilities of the methods described in the previous chapters. The UAVs do not have any remote sensing devices placed on them in order to minimize risk of instrumentation loss. All three UAVs are equipped with Pixhawk 4 flight controllers, GPS modules and Rasberry Pi microprocessors to enable autonomous navigation. The three UAVs are initialized and deployed in a sequential order with a 20 s lag between each takeoff. All UAVs concurrently fly in the sub-canopy airspace for approximately 10 minutes. Only one test is conducted because the sub-zero temperatures during the test shorten the battery life of each UAV. As discussed in the previous chapter, it is determined that the flight plan required an adequate buffer to ensure obstacle avoidance.

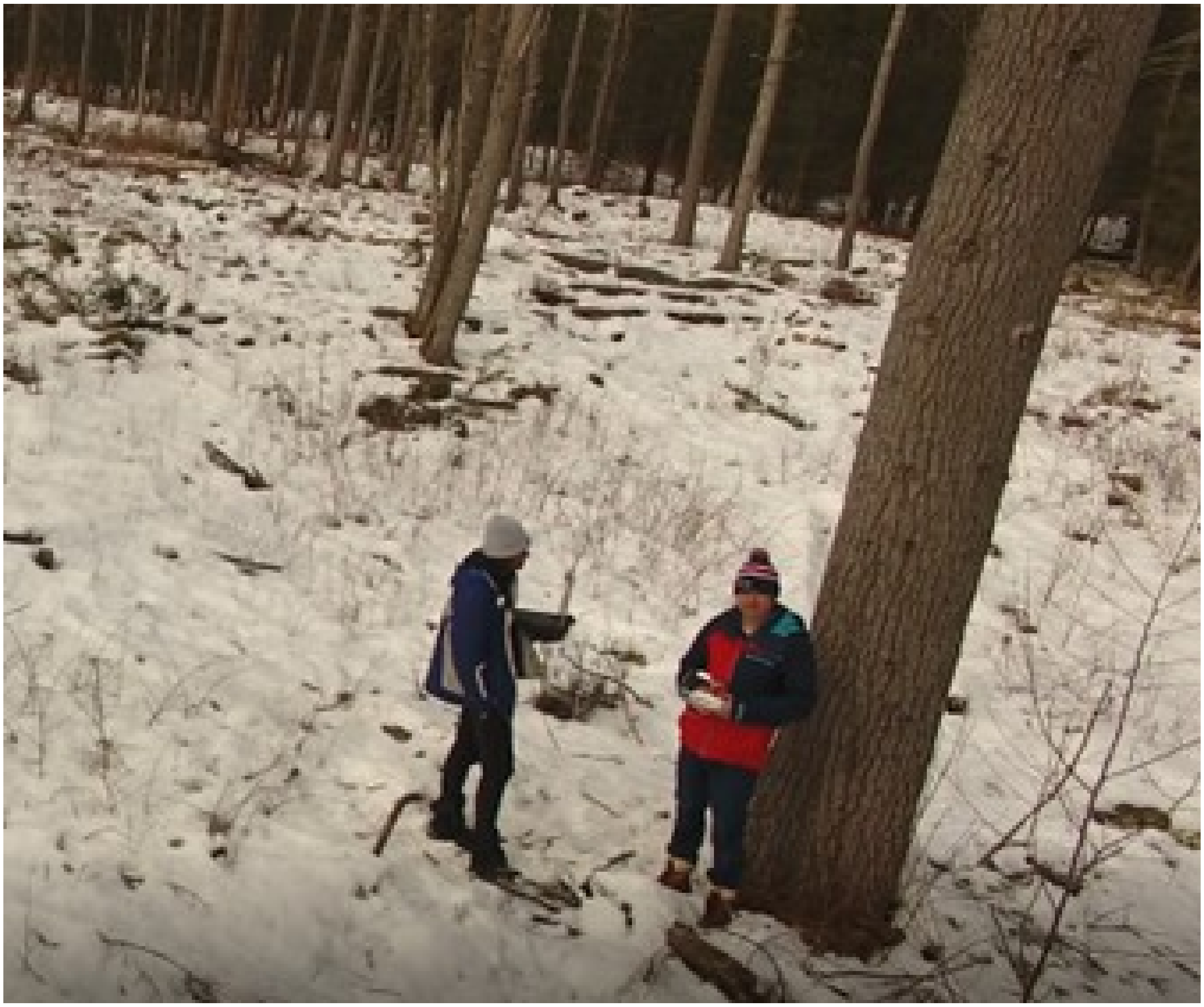


Figure 6.1: Kingman Farm Snow conditions During January to March 2023

The A* path planning algorithm is employed on the 2-10 m altitude band map after inflation of the obstacles by 2 m. The goal when creating the missions is to create a lawnmower style flight plan in the sub-canopy region. A multi-UAV mission is used to demonstrate the scalability of this path planning method and for more efficient forest coverage. A* is used to plan multiple connected paths for each UAV for the mission. The start and end points of each path is selected manually.

Before flying the test mission, a preliminary onsite walk through of Kingman Farm is conducted. Here, it is observed that the ground elevation varies significantly throughout the test site, although the altitude band maps show a relatively level ground surface with little elevation deviation. It is found later that the maximum elevation differential is 4.5 m. In addition, consideration is given to false barometer measurements during wind gusts which cause 1-2 m UAV altitude fluctuations. For these combined reasons, the flight altitude for the mission is chosen to be at 8 m within the 2-10 m altitude band. The flight speed is set to 2 m/s. The slow flight speed allows the user to observe the UAV platform throughout the mission. The planned flights and the recorded flights are shown in Figure 6.2.

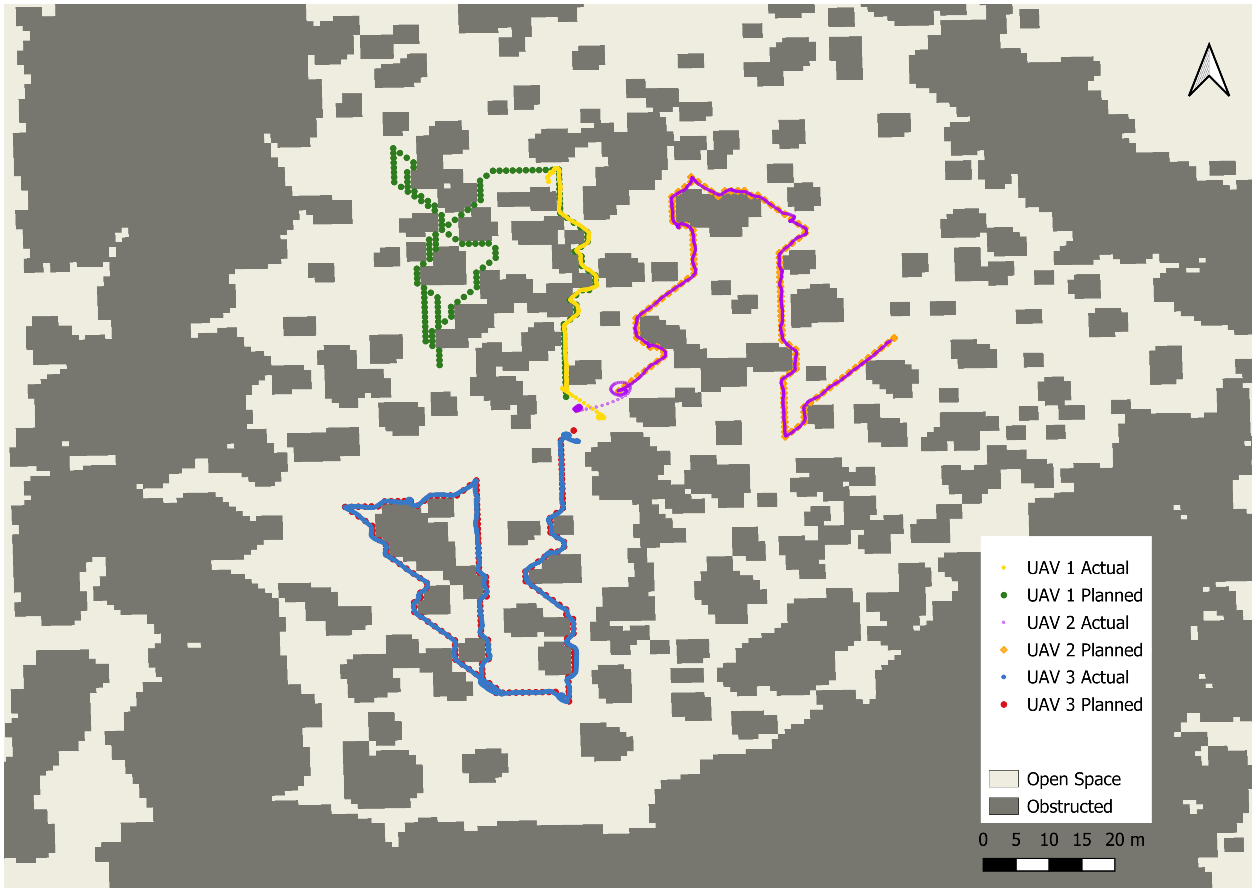


Figure 6.2: Actual and Planned missions for the three UAVs Sub-Canopy in the Kingman Farm 2-10m Altitude Band Map (March 2023). Obstructions are Trees

UAVs 1, 2, and 3 mission plans consisted of six, four, and five A* paths, respectively. The starting point of each UAV is chosen to be in the center of the forest. Because the UAVs are commanded remotely via a common ground station, close proximity is required, when the flight plan is initialized. The endpoints are also within close proximity of the initialization points in order to readily retrieve the UAVs. The start and end coordinates of each path of the UAVs can be found in Appendix B. The actual and planned paths are nearly identical. The maximum errors between the planned and actual paths of UAV 1, 2 and 3 are 0.87 m, 0.91 m and 0.88 m respectively.

UAV 1 does not complete the full planned path (Figure 6.2), as it encounters a stray branch that obstructed its path. Because this obstacle is not present on the LiDAR generated maps, avoidance is not possible. (The stray branch likely fell during one of the multiple snow fall events that took place between the LiDAR mapping of the area in November 2022 and the test flights in March 2023). When conducting sub-canopy flights, it is important to exercise caution if a significant amount of time has elapsed since the initial canopy mapping. Severe weather conditions should be monitored between these events that might cause changes in the forest conditions. Real-time obstacle detection sensors could be used to aid the UAV in the flight path tracking to make it robust. In addition, the planned paths shown in Figure 6.4 are always in open space pixels but the actual paths sometimes travels through obstructed pixels. This typically occurs when there are sharp turns in the mission plan. This is likely due to a combination of GPS error and UAV maneuvering limitations when performing sharp turns. The buffering of the obstacles account for this phenomena without hindering the UAV flights. The use of multiple UAVs is a clear advantage in this situation. The other two UAVs (UAV 2 and UAV 3) are able to successfully complete their respective missions, allowing for a large amount of data retrieval despite the loss of one UAV platform.

These tests show that the over-canopy LiDAR mapping and A* path planning methods, in conjunction with convoluted altitude-based obstacle maps, can be used to plan and execute sub-canopy multi-UAV navigation. Because GPS errors are the main concern, safety buffers are applied to ensure obstacle avoidance. While maps of trees are relatively consistent over time, modest changes in conditions occur that can result in collisions and failure. To limit these issues, the use of obstacle detection sensors in combination with these flight plans is recommended. The 8 m high elevation band is quite large and likely not viable for most forests. Alternatives for measuring altitude more precisely than the UAV's barometer would reduce that range.

CHAPTER 7

Photogrammetry Model Results from Sub-Canopy UAV Flights

7.1 Sub-Canopy Flight Plan

A DJI Phantom 4 UAV equipped with a 4K camera, mounted on a six axis gimbal, is used to produce 3D Structure from Motion (SfM) models. Snow-on data is collected in March 2023 and snow-off flights is collected in April 2023. Over-canopy and sub-canopy flights are conducted to compare the resulting point cloud model of the forest environment.

The above-canopy flights are conducted at 80 m above the forest floor in a typical lawn-mower style flight path. The camera for above canopy flights is directed downwards, perpendicular to the flight path. The camera for the sub-canopy flights is tilted 45° in relation to the flight path and aligned with the flight path. Sub-canopy flight paths target the middle section of search space (Figure 6.2). The start and end points of the path segments (as discussed in Chapter 6) are selected manually. The flight lines are generated using the A* path planning algorithm on the buffered 2 to 10 m altitude band map and are combined to create a single flight path. The flight is conducted at an 8 m altitude to account for altimeter readings during wind gusts that may cause a drop in the UAV altitude. Flying at the higher end of the elevation band tolerates UAV fluctuations within a safe range and enables the UAV to remain in open space.

7.2 Digital Surface Models (DSM)

SFM models are used to identify ground points. The model produces DSMs of the search area (Figure 7.1). The DSM, which is created by the over-canopy flight contains many “holes” or areas where no data is present (i.e, where snow depth is impossible to calculate). The results from the sub-canopy flight capture the entire surface. There is a considerable reduction of missing data pixels as compared to that of the above-canopy DSM.

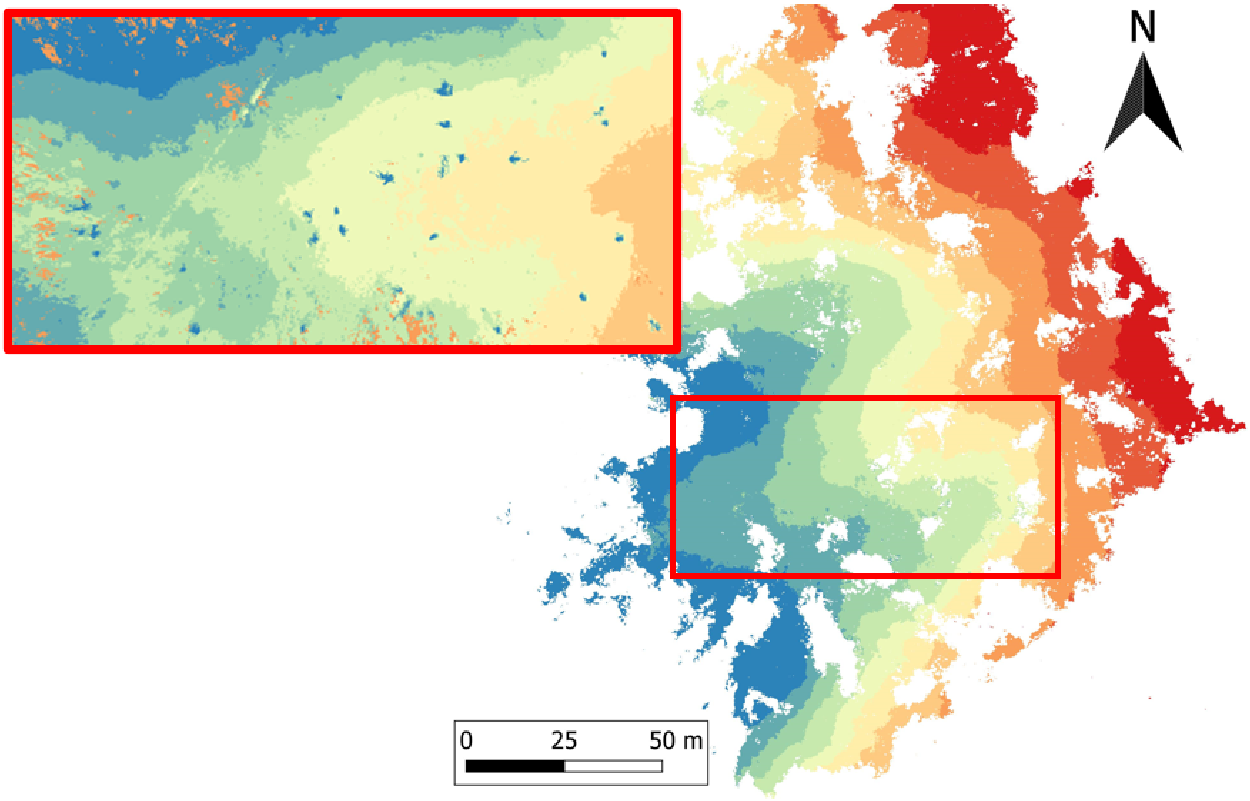


Figure 7.1: DSM Comparing Photogrammetry Results: (a) Below-Canopy and (b) Above-Canopy

The DSM from the above-canopy flights covers a larger area because the UAV altitude of 80 m allows a larger area to be observed by the camera. The sub-canopy flight DSM is smaller because the camera swath is reduced when flying 8 m above the forest floor. The white areas of the over-canopy DSM (Figure 7.1) are pixels where no data is produced. It is evident that the sub-canopy DSM has few white pixels. The improved observations from the sub-canopy DSM are an important step for snow depth measurement accuracy in the future.

7.3 3D SFM Models

Colorized 3D models are used to examine differences in map trees and surfaces under trees. The 3D SfM models (Figures 7.2 through 7.3) show significant improvement when produced using sub-canopy flights. For example, tree trunks are clearly visible in the sub-canopy models. This particular characteristic is important because snow under tree trunks will differ from those in open areas. Above-canopy flights have limited ability to capture the trees in these areas. The models in Figures 7.2 and 7.3 also show that the snow on the forest floor is clearly more visible in the sub-canopy flight models than the above-canopy models. The ground also has less discontinuities in the sub-canopy flight models as compared to that of the above-canopy flight models.

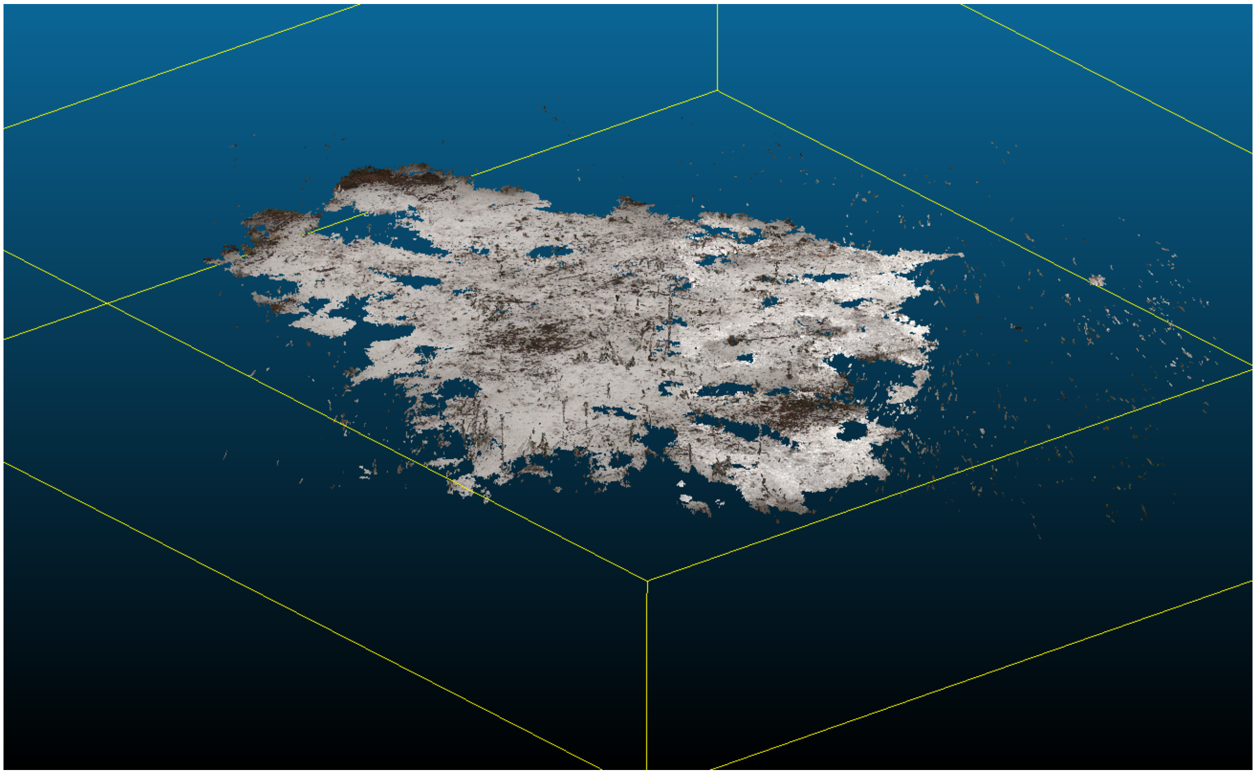


Figure 7.2: Above-Canopy 3D SfM RGB Model

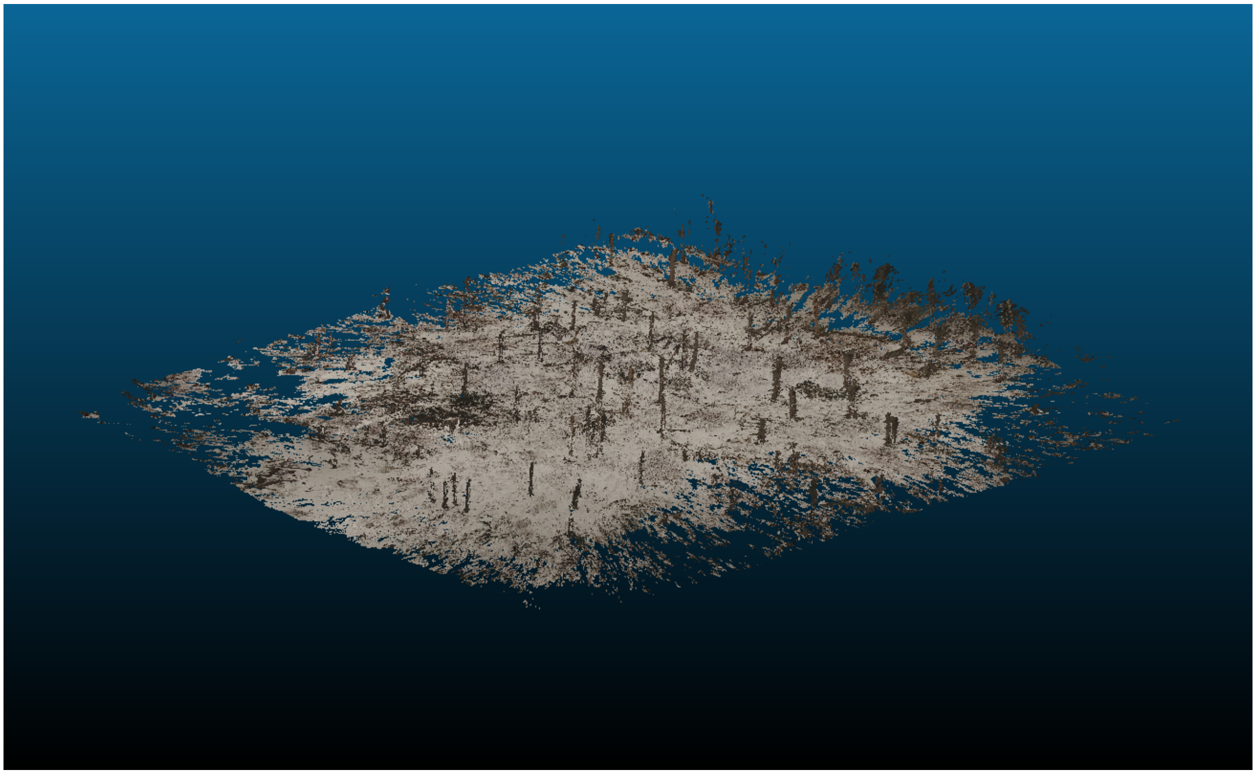


Figure 7.3: Sub-Canopy 3D SfM RGB Model

To examine the extent to which the ground can be discerned, transects showing the vertical profiles (Figures 7.4) are created from 3D models. The area around the bottom of the tree trunks is clearly defined for the sub-canopy transect. The above-canopy flights did not capture these areas because tree canopy obstructs the view by the UAV camera. The variation of the ground elevation is also clear and well defined by the sub-canopy model.

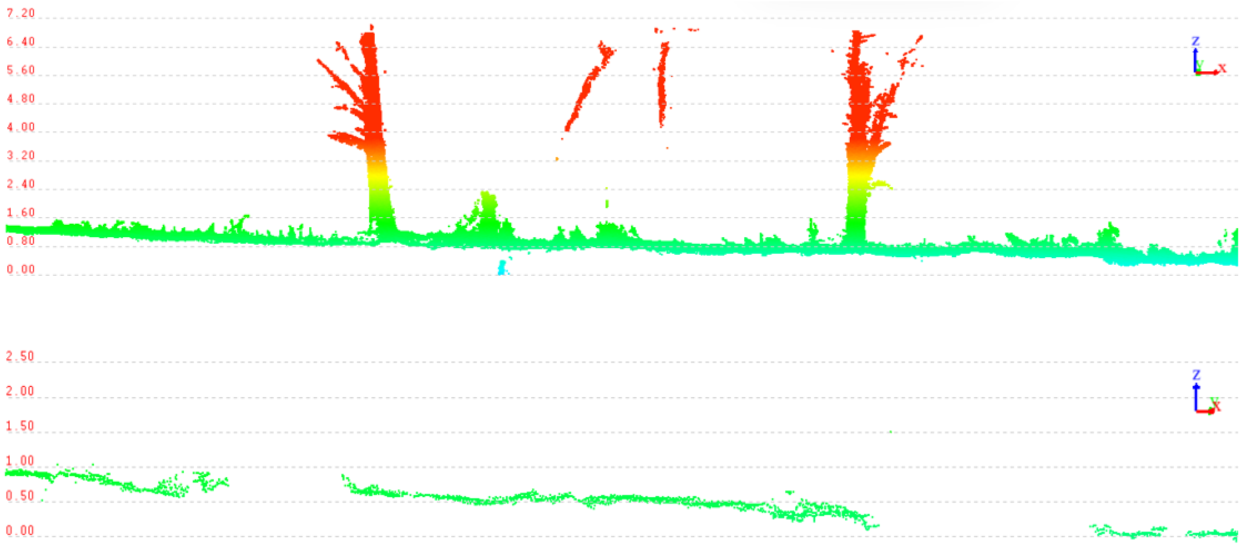


Figure 7.4: Altitude Profiles of SFM Models: Sub-Canopy Model (Top) and Above-Canopy Model (Bottom))

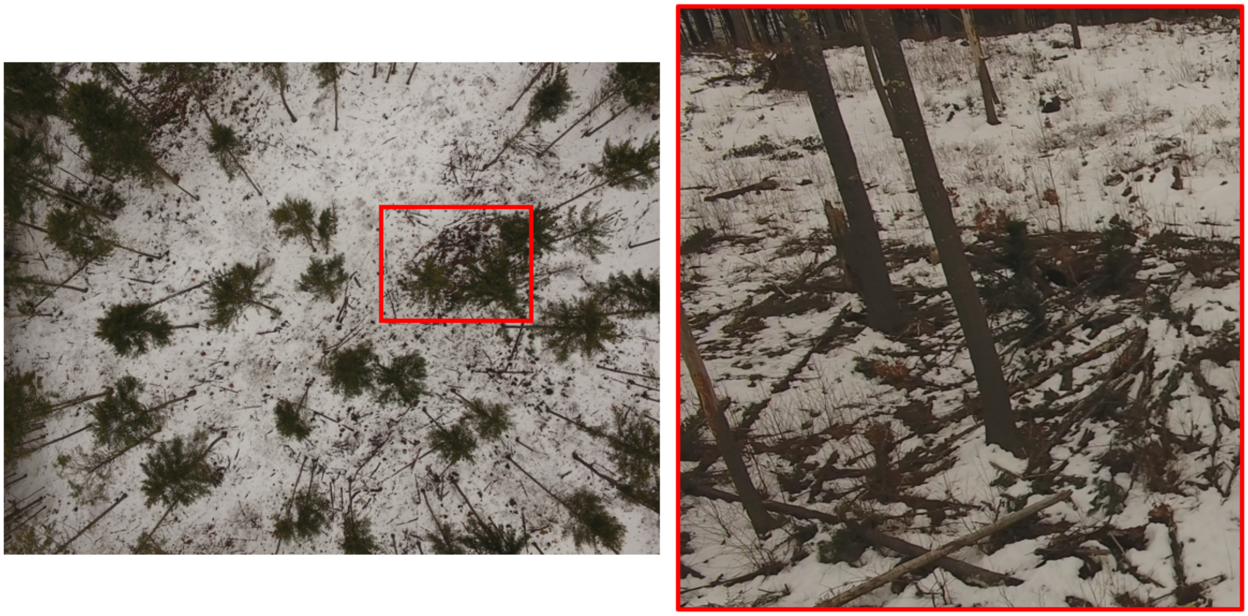


Figure 7.5: Above-Canopy UAV View (left) and Sub-Canopy UAV View (right) of the Same Area

The sub-canopy flight shows significant improvements in the models of the forest and can be used to more completely measure snow depth compared to that of over-canopy flights.

Overall, the sub-canopy flight shows a clear and complete ground model (Figure 7.5). This shows promise for creating continuous snow depth maps in this forest with the increased resolution of snow-on and snow-off DEMS from which snow depth can be calculated. The sub-canopy flights are also capable of creating 3D models of the forest environment which is not possible using over-canopy flights. Improvements that are needed include understanding which camera angle to use and the lighting conditions under which photogrammetry pictures are taken to measure snow depth. By flying below the obstructions of the canopy, the ability of sensors to collect data can be increased and help build a better understanding of forest environments. In addition to cameras, other sensors could be mounted on the UAV platform to improve data collection in forests. The methods described in this study also show that it is possible to safely navigate the UAV under the canopy and aid in producing the models.

CHAPTER 8

Conclusions

8.1 Results and Suggestions

Autonomous snow depth measurement resolutions are improved by leveraging currently existing LiDAR-based over-canopy flight technology. The LiDAR sensor is used to map sub-canopy obstacle layouts in 3D with high spatial resolution. The 3D models are processed to extract the 2D obstacles layout at a given height below the canopy. Altitudes with less obstructions are found by calculating the obstacle percentage of the 2D altitude specific obstacle maps. The A* path planning algorithm is used in this research to plan coverage mission paths in sub-canopy regions. Ox Bow Farm, Thompson Farm, and Kingman Farm are evaluated to find a suitable site to show proof-of-concept testing. Kingman Farm is found to have the sparsest sub-canopy environment followed by Thompson Farm and Ox Bow Farm. The method of using altitude band maps to plan A* paths is observed to perform best in sub-canopy environments where there is uniform tree growth and the obstacle density is between 20% and 42%. Ox Bow Farm and Thompson have denser growth (63%-85% and 27%-55% respectively). This restricts the free space along with the GPS satellite reception.

Multiple UAVs are flown in Kingman Farm, located in Madbury, New Hampshire. The results show that flight paths are tracked accurately. The maximum difference between A* produced setpoints and measured UAV trajectory is less than 1 m. Using LiDAR maps to enable autonomous navigation does not account for changes in the forest. Sub-canopy environment can change due to weather events and long periods of time between the LiDAR mapping and sub-canopy flights. During the final test, one UAV platform experienced catastrophic failure due to a stray branch in the mission path. For this reason, weather events should be monitored between above-canopy mapping and below-canopy flights. The time between these events should be minimised and altitude band maps should be updated between seasons.

The proposed methods will not be successful in very dense under canopy environments. The two main reasons that successful navigation cannot be achieved is a lack of GPS satellites for navigation and insufficient free space for UAVs to navigate. The obstacle density is measured for three tests sites (Ox Bow Farm, Kingman Farm, Thompson Farm). The sparsest site is chosen to show proof-of-concept. Obstacle sizes are artificially increased through image convolution in order to account for GPS location errors during UAV navigation. An altitude band of 2-10 m is used to account for errors in the barometer used to measure altitude and for variations in ground elevation. The flight altitude is chosen to ensure that the UAV is not within proximity of the thick undergrowth of the forest.

Above-canopy flights cannot be used to generate SFM models for snow depth calculation because of the obstructions in the camera view. By flying below-canopy, more detail can be discerned and accurate DEMs are developed. Photogrammetry-based SFM models exhibit increased point cloud ground return density when comparing sub-canopy to above-canopy flights. This shows promise for increasing the resolution of snow depth data and the ability to use SFM based methods to detect sub-canopy snow depths.

8.2 Future Work

For this proof-of-concept testing, an altitude band of 2-10 m is chosen. Because such a large band is chosen, more objects in the vertical forest column are included as obstructions. This inclusion further limits the free space in which the UAV is allowed to travel. However, by using terrain-following (i.e., single-point) LiDAR, the UAV is able to maintain altitude within a more appropriately compact band, instead of having to buffer the band because of barometer fluctuations and ground elevation variability. From observations of altitude band maps, it is evident that varying forest areas have free space in different altitudes. As such, it would be beneficial to expand UAV path planning to make use of the full 3D space, taking advantage of free space in multiple altitude bands. It should be noted, however, that the A* method is not well-suited for 3D path planning (referred to as "the curse of dimensionality") and, thus, research into other path planning methods would be necessary.

Further consideration of camera mounting angles, as well as flight paths (other than the traditional lawnmower coverage path), could improve SFM models in both observational perspectives (i.e., from various angles) and in revealing otherwise hidden areas due to obstructions. In addition, lighting is critical to the accuracy of the SFM model. Therefore, for future missions, it would be prudent to understand how natural lighting conditions affect the accuracy of RGB representation of snow on the forest floor. From these observations, the user would be able to determine the optimum times of the day during which to fly such sub-canopy missions.

Although video data is compiled as a post-processing exercise, the data from video occupies significant onboard data storage, especially when using high-resolution cameras, and requires considerable processing power to obtain satisfactory SFM models. At the same time, many frames are redundant (i.e., show the same view) without the benefit of additional information. Further investigation into video sampling, therefore, could minimize onboard data storage requirements as well as decreasing post processing efforts.

Finally, the integration of over-canopy and sub-canopy UAV systems would benefit this technology. The areas where over-canopy flight cannot achieve adequate snow depth resolution can be identified and sub-canopy UAVs can be alerted to survey those regions. This would make the process efficient and seamless.

LIST OF REFERENCES

[1] Simon Ecke, Jan Dempewolf, Julian Frey, Andreas Schwaller, Ewald Endres, Hans-Joachim Klemmt, Dirk Tiede, and Thomas Seifert. UAV-Based Forest Health Monitoring: A Systematic Review. *Remote Sensing*, 14(13):3205, January 2022. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

[2] https://www.facebook.com/akashhere17. A* Algorithm | Introduction to the A* Seach Algorithm, December 2019. Section: Uncategorized.

[3] PX4 System Architecture | PX4 User Guide.

[4] OAR US EPA. Climate Change Indicators: Snow Cover, July 2016.

[5] Thomas H. Painter, Felix C. Seidel, Ann C. Bryant, S. McKenzie Skiles, and Karl Rittger. Imaging spectroscopy of albedo and radiative forcing by light-absorbing impurities in mountain snow. *Journal of Geophysical Research: Atmospheres*, 118(17):9511–9523, 2013. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jgrd.50520.

[6] JONATHAN P. WOLFE and JEFFERSON R. SNIDER. A Relationship between Reflectivity and Snow Rate for a High-Altitude S-Band Radar in: Journal of Applied Meteorology and Climatology Volume 51 Issue 6 (2012).

[7] Roy Rasmussen, Michael Dixon, Steve Vasiloff, Frank Hage, Shelly Knight, J. Vivekanandan, and Mei Xu. Snow Nowcasting Using a Real-Time Correlation of Radar Reflectivity with Snow Gauge Accumulation. *Journal of Applied Meteorology and Climatology*, 42(1):20–36, January 2003. Publisher: American Meteorological Society Section: Journal of Applied Meteorology and Climatology.

[8] Raphael Zon and William Norwood Sparhawk. *Forest Resources of the World*. McGraw-Hill Book Company, Incorporated, 1923. Google-Books-ID: esEHAQAAIAAJ.

[9] Andrés Varhola, Nicholas C. Coops, Markus Weiler, and R. Dan Moore. Forest canopy effects on snow accumulation and ablation: An integrative review of empirical results. *Journal of Hydrology*, 392(3-4):219–233, October 2010.

[10] Jennifer M. Jacobs, Adam G. Hunsaker, Franklin B. Sullivan, Michael Palace, Elizabeth A. Burakowski, Christina Herrick, and Eunsang Cho. Snow depth mapping with unpiloted aerial system lidar observations: a case study in Durham, New Hampshire,

United States. *The Cryosphere*, 15(3):1485–1500, March 2021. Publisher: Copernicus GmbH.

[11] Marlene L. Huerta, Noah P. Molotch, and James McPhee. Snowfall interception in a deciduous Nothofagus forest and implications for spatial snowpack distribution. *Hydrological Processes*, 33(13):1818–1834, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hyp.13439.

[12] R. A. Schmidt. *Sublimation of Wind-transported Snow: A Model.* Rocky Mountain Forest and Range Experiment Station, Forest Service, U.S. Department of Agriculture, 1972. Google-Books-ID: ndB586FaY2UC.

[13] Pierre Y. Bernier. Extrapolating Snow Measurements on the Marmot Creek Experimental Basin. *Hydrology Research*, 17(2):83–92, April 1986.

[14] HUI XU, J. O. BAILEY, E. C. BARRETT, and R. E. J. KELLY. Monitoring snow area and depth with integration of remote sensing and GIS. *International Journal of Remote Sensing*, 14(17):3259–3268, November 1993. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01431169308904440.

[15] R. Marti, S. Gascoin, E. Berthier, M. de Pinel, T. Houet, and D. Laffly. Mapping snow depth in open alpine terrain from stereo satellite imagery. *The Cryosphere*, 10(4):1361–1380, July 2016.

[16] Yuqiong Liu, Christa D. Peters-Lidard, Sujay Kumar, James L. Foster, Michael Shaw, Yudong Tian, and Gregory M. Fall. Assimilating satellite-based snow depth and snow cover products for improving snow predictions in Alaska. *Advances in Water Resources*, 54:208–227, April 2013.

[17] Hans Lievens, Isis Brangers, Hans-Peter Marshall, Tobias Jonas, Marc Olefs, and Gabriëlle De Lannoy. Sentinel-1 snow depth retrieval at sub-kilometer resolution over the European Alps. *The Cryosphere*, 16(1):159–177, January 2022. Publisher: Copernicus GmbH.

[18] Per-Erik Mellander, Hjalmar Laudon, and Kevin Bishop. Modelling variability of snow depths and soil temperatures in Scots pine stands. *Agricultural and Forest Meteorology*, 133(1):109–118, November 2005.

[19] Giulia Mazzotti, William Ryan Currier, Jeffrey S. Deems, Justin M. Pflug, Jessica D. Lundquist, and Tobias Jonas. Revisiting Snow Cover Variability and Canopy Structure Within Forest Stands: Insights From Airborne Lidar Data. *Water Resources Research*, 55(7):6198–6216, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2019WR024898.

[20] M. Nolan, C. Larsen, and M. Sturm. Mapping snow depth from manned aircraft on landscape scales at centimeter resolution using structure-from-motion photogrammetry. *The Cryosphere*, 9(4):1445–1463, August 2015. Publisher: Copernicus GmbH.

[21] Joachim Meyer and S. McKenzie Skiles. Assessing the Ability of Structure From Motion to Map High-Resolution Snow Surface Elevations in Complex Terrain: A Case Study From Senator Beck Basin, CO. *Water Resources Research*, 55(8):6596–6605, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2018WR024518.

[22] Salvatore Manfreda, Matthew F. McCabe, Pauline E. Miller, Richard Lucas, Victor Pajuelo Madrigal, Giorgos Mallinis, Eyal Ben Dor, David Helman, Lyndon Estes, Giuseppe Ciraolo, Jana Müllerová, Flavia Tauro, M. Isabel De Lima, João L. M. P. De Lima, Antonino Maltese, Felix Frances, Kelly Caylor, Marko Kohv, Matthew Perks, Guiomar Ruiz-Pérez, Zhongbo Su, Giulia Vico, and Brigitta Toth. On the Use of Unmanned Aerial Systems for Environmental Monitoring. *Remote Sensing*, 10(4):641, April 2018. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[23] Telmo Adão, Jonáš Hruška, Luís Pádua, José Bessa, Emanuel Peres, Raul Morais, and Joaquim João Sousa. Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry. *Remote Sensing*, 9(11):1110, November 2017. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[24] Maarten Uijt de Haag, Chris G. Bartone, and Michael S. Braasch. Flight-test evaluation of small form-factor LiDAR and radar sensors for sUAS detect-and-avoid applications. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–11, September 2016. ISSN: 2155-7209.

[25] Ryan A. Chisholm, Jinqiang Cui, Shawn K. Y. Lum, and Ben M. Chen. UAV LiDAR for below-canopy forest surveys. *Journal of Unmanned Vehicle Systems*, 01(01):61–68, December 2013.

[26] Holybro S500 ARF V2- 480mm Quadcopter Platform - Frame Kit.

[27] https://www.facebook.com/lifewire. DJI Phantom 4 Pro V. 2.0 Review. Section: Lifewire.

[28] Daniel Foead, Alifio Ghifari, Marchel Budi Kusuma, Novita Hanafiah, and Eric Gunawan. A Systematic Literature Review of A* Pathfinding. *Procedia Computer Science*, 179:507–514, January 2021.

[29] Y Y M Leong. Australian Research Centre for Aerospace Automation. page 21.

[30] Autonomous UAV Control and Mapping in Cluttered Outdoor Environments...

[31] Tao Zheng, Yanqiang Xu, and Da Zheng. AGV Path Planning based on Improved A-star Algorithm. In *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 1534–1538, October 2019.

[32] C. J. Perron, K. Bennett, and T. D. Lee. Forest stewardship plan: Thompson farm, nh. Technical report, University of New Hampshire, Produced by Ossipee Mountain Land Company, West Ossipee, 2004.

[33] GreenValley International. Lidar technology. https://greenvalleyintl.com/static/upload/file/20210817/1629190611196646.pdf, 2021. Accessed on May 9, 2023.

[34] Atheer L Salih, M Moghavvemi, Haider A F Mohamed, and Khalaf Sallom Gaeid. Flight PID controller design for a UAV quadrotor.

[35] Controller Diagrams | PX4 User Guide.

[36] Dario Brescianini, Markus Hehn, and Raffaello D'Andrea. Nonlinear Quadrocopter Attitude Control: Technical Report. Technical report, ETH Zurich, 2013. Medium: application/pdf,Online-Ressource.

APPENDIX A

Quadcopter Dynamics and Controls

A.1 General Dynamics

The quad rotor X configuration is one of the most popular platforms for uncrewed aerial vehicle systems because it is very versatile and can be made compact for a variety of applications including search and rescue, surveys, and transport [citation]. The platform consists of a rigid body connected to 4 arms with rotors at the end. The arms sit in a X configuration centered around the rigid body. With these 4 rotors the quadcopter can move laterally in the X, Y, and Z directions as well as body rotations of roll pitch and yaw. The rotor speeds are denoted by ω_i and by manipulating the ω of each rotor different maneuverability configurations can be achieved. The UAV has 6 states that are controlled by the 4 motor inputs. The 6 states include translational motion in the X,Y, Z earth frame axes and roll pitch yaw [Φ, ϑ, Ψ] which are the Euler rates. Roll corresponds to rotation about the x axis, pitch is rotation about the y axis and yaw is the rotation about the Z axis.

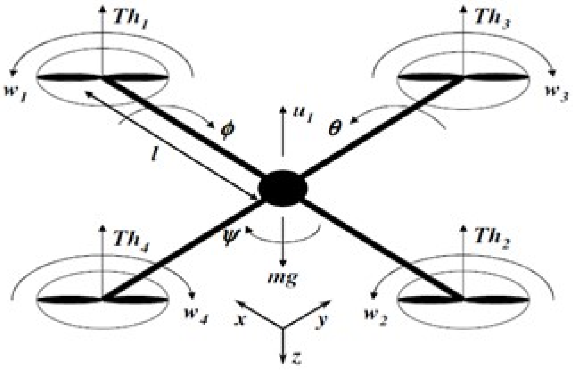


Figure A.1: Quadrotor dynamics

The quadcopter platform is an underactuated system because it has 6 degrees of freedom and only 4 actuators to control those states, because of this, modelling a quadrotor aircraft is difficult as the states are all coupled, and the actuator inputs can affect multiple states at once. The goal is to develop a model which closely describes the dynamics.

The quad rotor has four rotors which are attached to the body of the platform with 4 equal length arms and these rotors provide 4 different thrusts. The sum of these thrusts is what provide the control inputs necessary to maneuver the UAV in the Z direction. The rotors produce moments about the three Cartesian coordinates which allow for maneuverability in the X and Y directions. The roll moment helps provide thrusts along the Y axis and the pitch moment allows for thrusts to be directed along the X axis. To achieve movement along the Z direction, the moments produced by the rotors must be canceled which is achieved by spinning pairs of rotors in opposing directions with equal magnitude. Pitch is achieved by spinning the front and back rotors at different speeds. Roll is achieved by spinning the side rotors at different speeds. Of the four rotors, two must spin counterclockwise and the other two must spin clockwise so that moments can be created in all directions while maintaining an overall thrust upwards. The last degree of freedom left is the yaw state which can be controlled by spinning the rotors that lie at the end of the arms of the X configuration at different speeds. Because the system is under actuated, the 6 states cannot be controlled simultaneously so the yaw moment is necessary to decouple the states for easier maneuverability and reduced time between controlling the states [34].

The combinations of rotor speeds which achieve upward thrust, roll, pitch, and yaw can be compounded and defined as u_i and these can be the 4 control inputs to the system.

$$u_1 = k_f \left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \tag{A.1}$$

$$u_2 = k_f \left(\omega_4^2 - \omega_2^2\right) \tag{A.2}$$

$$u_3 = k_f \left(-\omega_3^2 + \omega_1^2\right) \tag{A.3}$$

$$u_4 = k_M \left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \tag{A.4}$$

Here u_1 u_2 u_3 u_4 are total upward thrust, roll, pitch, and yaw respectively. kM is the moment of inertia coefficient measured in $(N*m*s^2)$ and kf is the thrust coefficient measured in $(N*s^2)$.

The state variables can be modelled using the following equations:

$$\ddot{x} = \frac{-1}{m}\left[k_{tx}x' + u_1\left(\sin\Phi\sin\Psi + \cos\Phi\cos\Psi\sin\theta\right)\right] \tag{A.5}$$

$$\ddot{y} = \frac{-1}{m}\left[k_{ty}y' + u_1\left(\sin\theta\cos\Psi + \cos\Phi\cos\Psi\sin\theta\right)\right] \tag{A.6}$$

$$\ddot{z} = \frac{-1}{m}\left[k_{t_z}\dot{z} - mg + u_1\left(\cos\Phi\cos\theta\right)\right] \tag{A.7}$$

$$\ddot{p} = \frac{-1}{I_x}\left[k_{rx}p - lu_2 - I_y qr + I_z qr + I_r q\omega_r\right] \tag{A.8}$$

$$\ddot{q} = \frac{-1}{I_y}\left[k_{ry}q - lu_3 - I_x pr + I_z pr + I_r p\omega_r\right] \tag{A.9}$$

$$\ddot{r} = \frac{-1}{I_z}\left[u_4 - k_{r_z} r + I_x pq - I_y pq\right] \tag{A.10}$$

I $(kg * m^2)$ is the moment of inertia of the UAV, l (m) is the length of the UAV arms, $I_r(kg * m^2)$ is the moment of inertia of the motors, m (kg) is the mass of the UAV, g (m/s^2) is the acceleration due to gravity, k_t $(N * s/m)$ is the aerodynamic thrust drag coefficient, and k_r $(N * m/s)$ is the aerodynamic moment drag coefficient. The relationship between the inertial angular rates to the Euler rates are defined as follows:

$$\ddot{\Phi} = \ p + r\cos\Phi\tan\theta + q\sin\Phi\tan\theta \tag{A.11}$$

$$\ddot{\theta} = \ q\cos\Phi - r\sin\Phi \tag{A.12}$$

$$\ddot{\Psi} = r\frac{\cos\Phi}{\tan\theta} + q\frac{\sin\Phi}{\cos\theta} \tag{A.13}$$

From the above equations it is apparent that the quadcopter system is a highly coupled nonlinear system. A PID control scheme can be used to command the UAV because the model can be disregarded and the error between current states and setpoints can be measured. If the digital controller has a fast enough response rate, adjustments can be made to the motor inputs to alter the overall thrusts and achieve exact angular attitudes and positions.

A.2 General Control System

Multiple PID controllers are utilized to make sure each state can be commanded with desirable response times. One controller is used the control the inertial frame of the UAV which includes the X, Y, and Z positions and velocities and the other controller is responsible for the Body frame of the UAV which includes the Euler angles and rates.

To achieve the required performance, a PX4 autopilot system is used, which employs a sensor feedback based PID control scheme to command the UAV. The Pixhawk is a digital controller and further information can be found in the above chapters. In this section a detailed discussion will be provided on the control scheme used by the Pixhawk and simulations will be used to verify the validity of the system.

The overall schematic for the Pixhawk controller is subdivided into the inertial frame controller and the body frame controller. The connection between the two is facilitated through a converter which transforms the acceleration setpoints to attitude setpoints using the equations of motion described above.

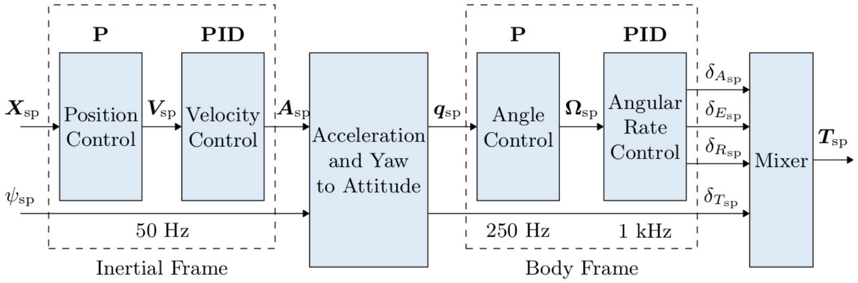


Figure A.2: Overall Px4 Control Scheme

A.3 Inertial Frame Controllers

A.3.1 Inertial Position Controller

The positions, and Euler angles are controlled using a P controller and the corresponding rates are controlled using a PID controller. The communication rates used by the system are also listed above and these are the frequencies at which sensor feedback and control signal output is necessary to maintain control over the UAV platform.

An assumption that needs to be made for the cascaded control scheme shown above is that the inner loop (the body frame) controller needs to respond significantly faster than the outer loop (the inertial frame) controller which can be seen by the communication rates.

To understand the control structure, each individual controller needs to be analyzed starting with the position controller.

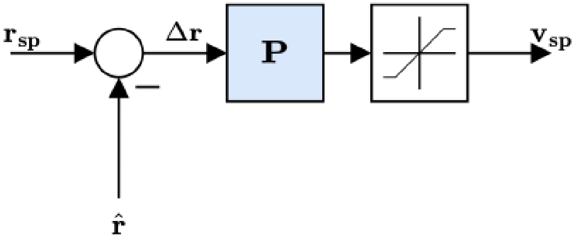


Figure A.3: Position Px4 Control Scheme

The structure of the inertial frame position controller consists of a feedback path where the GPS sensor provides the controller with latitude and longitude positions and a barometer provides altitude measurements ($\hat{r}$). Δr is the error term calculated by measuring the difference between the setpoints (r_{sp}) being given to the controller. A proportional gain P is multiplied with the error term which produces a velocity command for the velocity controller.

A.3.2 Inertial Velocity Controller

The velocity controller has the following structure:

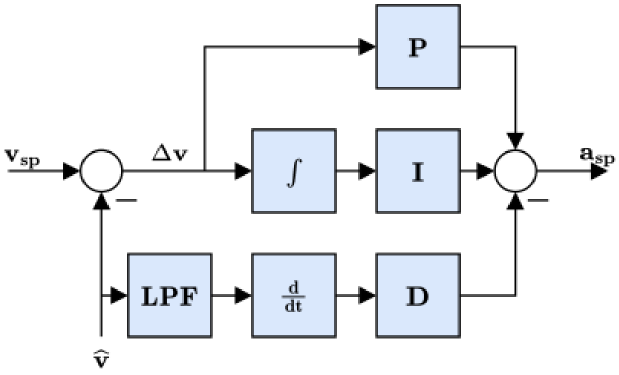


Figure A.4: Velocity Px4 Control Scheme

The velocity setpoint ($\hat{v}$) is obtained by integrating the Inertial Measurement Units (IMU) signal for the linear accelerations in the X, Y, and Z directions. The velocity setpoint (v_{sp}) is the input from the position controller above and a new error term is calculated (Δv). The velocity measurement incoming from the sensor is also sent through a low-pass filter (LPF) to reduce high frequency noise before differentiating the signal. By doing this, the controller can ensure rejection of sensor noise. The differentiated signal is multiplied with a derivative gain D before being added to the rest of the outputs from the P and I portion of the controller. The integral gain has anti reset windup (ARW) implemented which allows for saturation limits of the actuators to not be reached [35]. ARW is especially helpful in dynamic systems that are constantly subjected to the force of gravity such as the UAV when in flight. The ARW system allows for the Pixhawk to reject buildup of the integral term which in turn helps protect the motors from being over stressed when being commanded.

$$u_i = \begin{cases} M^+ & \text{if } u_i \ > \ M^+ \\ M^+ & \text{if } u_i \ < \ M^- \\ u_i & \text{if otherwise} \end{cases} \tag{A.14}$$

Here u_i is the commanded output and M^+ and M^- are saturation limits of the motors being used on the UAV. The saturation limit is not checked during the output of the velocity controller, instead it is applied to the signal at the final output of the body frame controller to the motor voltages.

The output of the integrated signal is multiplied with the I gain and added to the P and D gain signal to produce the velocity setpoints that can be used by the body frame controller.

For the UAV to move laterally in the inertial frame, the attitude of the aircraft needs to be manipulated. Therefore, the acceleration setpoints that are outputted from the inertial frame controller are mapped to the necessary attitudes using the equations of motion described above. The calculated attitudes (roll, pitch, and yaw) are the inputs for the body frame controller. The body frame controller has an angular position controller and an angular rate controller and the final output from the rate controller are the motor voltages that produce the necessary thrusts to move the UAV in the inertial frame.

A.4 Body Frame Controllers

A.4.1 Body Frame Attitude Controller

The angular position controller only provides a proportional (P) gain and has the following structure:

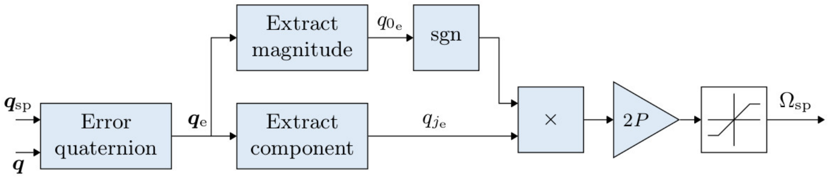


Figure A.5: attitude Px4 Control Scheme

The q_{sp} is the final output of the converter between inertial frame accelerations and body frame orientations. q is the measured orientation that is produced by integrating the gyroscope sensor signal.

The orientation of the UAV platform can be described using three basic rotations around the X, Y and Z axes centered around the body of the UAV platform. The three elementary rotations are known as the Euler angles and the sequence is important when calculating the orientation. A popular aircraft rotation sequence is the ZYX scheme [36].

1. Initial rotation about the Z axis in the inertial frame. (Ψ yaw)

2. Rotation about the new Y axis. (ϑ pitch)

3. Rotation about the new X axis. (Φ roll)

The sequence can be described using matrices in the following manner:

$$R_{BI} = R_{BL}R_{LK}R_{KI} \tag{A.15}$$

$$R_{BI} = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\Phi & s_\Phi \\ 0 & -s_\Phi & c_\Phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\Psi & s_\Psi & 0 \\ -s_\Psi & c_\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\right]$$

$$R_{BI} = \left[\begin{bmatrix} c_\theta c_\Psi & c_\theta s_\Psi & -s_\theta \\ -c_\theta s_\Psi \ + s_\theta s_\Phi c_\Psi & c_\Phi c_\Psi \ + s_\theta s_\Phi s_\Psi & s_\Phi c_\theta \\ s_\Phi s_\Psi \ + s_\theta c_\Phi c_\Psi & -s_\Phi s_\Psi \ + s_\theta c_\Phi c_\Psi & c_\Phi c_\theta \end{bmatrix}\right]$$

Here (c) and (s) represent the cosine and sine function respectively.

The body frame controller utilizes quaternions to make control calculations quicker and lessen the load on the processor. The sequence of three rotations can be parameterized by a single rotation about a fixed axis which has directional components $\vec{k}$ and rotation angle α. This single rotation is described using a quaternion and reduces the computational load because now there is only one rotation instead of three separate ones. The quaternion is formally described as follows:

$$\begin{bmatrix} q = [\ q_0\ q_1\ q_2\ q_3]^T = \begin{bmatrix} q_0 \\ q_{1:3} \end{bmatrix} = \begin{bmatrix} cos(\frac{\alpha}{2}) \\ \vec{k}sin(\frac{\alpha}{2}) \end{bmatrix} \end{bmatrix} \tag{A.16}$$

The four components of the quaternion must satisfy the following constraint:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{A.17}$$

The adjoint, norm and inverse of the quaternion q are defined as follows:

$$\bar{q} = \begin{bmatrix} q_0 \\ -q_{1:3} \end{bmatrix} \tag{A.18}$$

$$||q|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{A.19}$$

$$q^{-1} = \frac{\bar{q}}{||q||} \tag{A.20}$$

Therefore, the goal is to create an error quaternion between the desired attitude quaternion and the current attitude quaternions and apply a proportional gain to produce the desired body rate $\vec{\Omega}$.

The quadcopter dynamics can be modelled in the following way:

$$\dot{q} = \begin{bmatrix} \dot{q_0} \\ \dot{q_{1:3}} \end{bmatrix} = \frac{1}{2}q \bigotimes p(\vec{\Omega}) = \begin{bmatrix} -\frac{1}{2}q_{1:3}^T\vec{\Omega} \\ \frac{1}{2}(S(q_{1:3}) + q_0 I)\vec{\Omega} \end{bmatrix} \tag{A.21}$$

With $S(q_{1:3})$ denoting the skew matrix of $q_{1:3}$:

$$S(q_{1:3}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{A.22}$$

Here q is the current attitude of the platform, and $p(\vec{\Omega})$ is the quaternion representation of the angular body rates. Therefore, if the $p(\vec{\Omega})$ can be calculated using the attitude controller, a command can be sent to the angular rate controller to achieve the desired rates. Let us also consider the fact that any angular attitude can be achieved by spinning clockwise or counterclockwise about the quaternion defined rotation axis. This can sometimes lead to unnecessary full rotations and to solve this problem the controller must satisfy the following condition:

$$q(\vec{\Omega}_{cmd}) = -q((\vec{\Omega}_{cmd}) \tag{A.23}$$

Where $(\vec{\Omega}_{cmd})$ is the commanded angular rates. This condition can be achieved by making the sign of q negative whenever $q_0 < 0$. Therefore, the following control law is used by the Pixhawk flight controller:

$$q\left(\vec{\Omega}_{cmd}\right) = 2P * \left[sgn\left(q_{e,0}\right)\right] q_{e,1:3} \tag{A.24}$$

Where P is the proportional gain provided by the P controller, q_e is the error quaternion representing the difference between the desired attitude q_{cmd} and the current attitude q, and sgn is defined in the following manner:

$$sgn(q_{e,0}) = f\left(x\right) = \begin{cases} 1 & \text{if } q_{e,0} \geq\ 0 \\ -1 & \text{if } q_{e,0} <\ 0 \end{cases} \tag{A.25}$$

$$q_e = q^{-1} \bigotimes q_{cmd} \tag{A.26}$$

The commanded angular rates are calculated using this control law. Inserting the commanded angular rates into the equation for the model of the system describes how the orientation changes with time and results in the following:

$$\dot{q} = \frac{1}{2} q \bigotimes p(\vec{\Omega}_{cmd}) \tag{A.27}$$

The final output of the attitude controller are the commanded angular rates or the setpoints necessary to achieve the desired output.

$$\vec{\Omega}_{cmd} = \vec{\Omega}_{sp} \tag{A.28}$$

A.4.2 Body Frame Angular Rate Controller

The above calculated angular rates are the inputs for the angular rate controller and the system has the following structure:

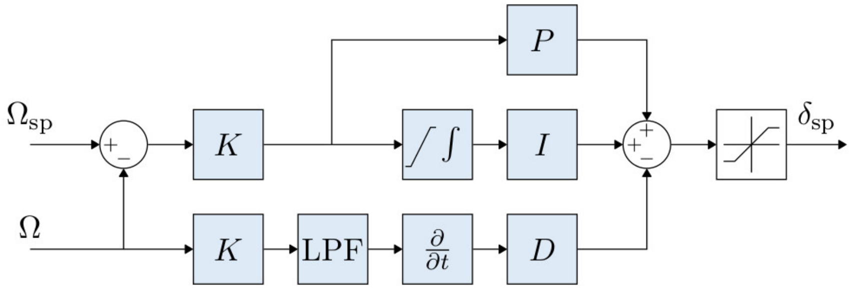


Figure A.6: attitude rate Px4 Control Scheme

The Pixhawk firmware has two versions of this controller implemented in the firmware. Both versions are mathematically equivalent and known as the parallel or standard form.

The user can switch between the types of rate controller by setting K=1 which induces the parallel form or P=1 which enables the standard form.

Standard form:

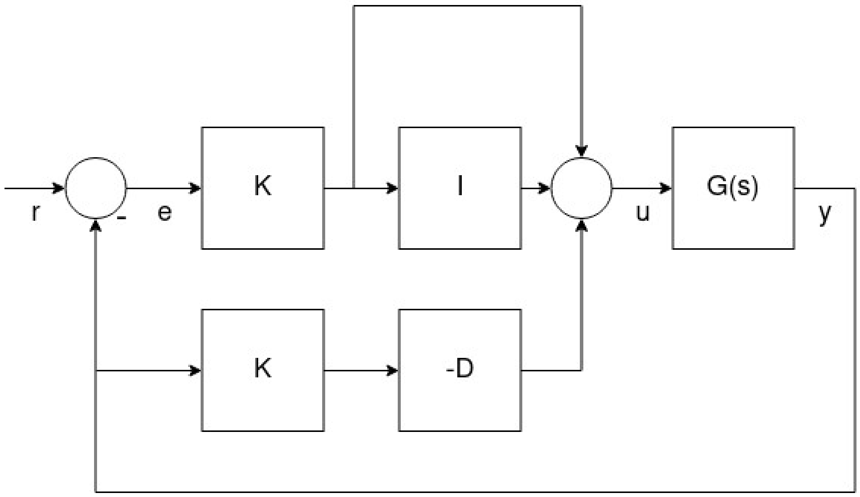


Figure A.7: Standard Attitude Rate Px4 Control Scheme

The main benefit to this form is that it decouples the proportional gain tuning from the integral and derivative gains. This allows easier tuning for freshly built platforms because the derivative and integral gains can be implemented from similarly designed UAV and the K gain can be adjusted independently till desired flying behavior is achieved.

Here G(s) represents the quadcopter, r is the setpoints calculated by the attitude controller, y is the gyroscope measured angular rates of the UAV, e is the error between the setpoints and the measured rates, and u is the output of the controller which in this case the angular velocities of the 4 rotors.

Parallel Form:

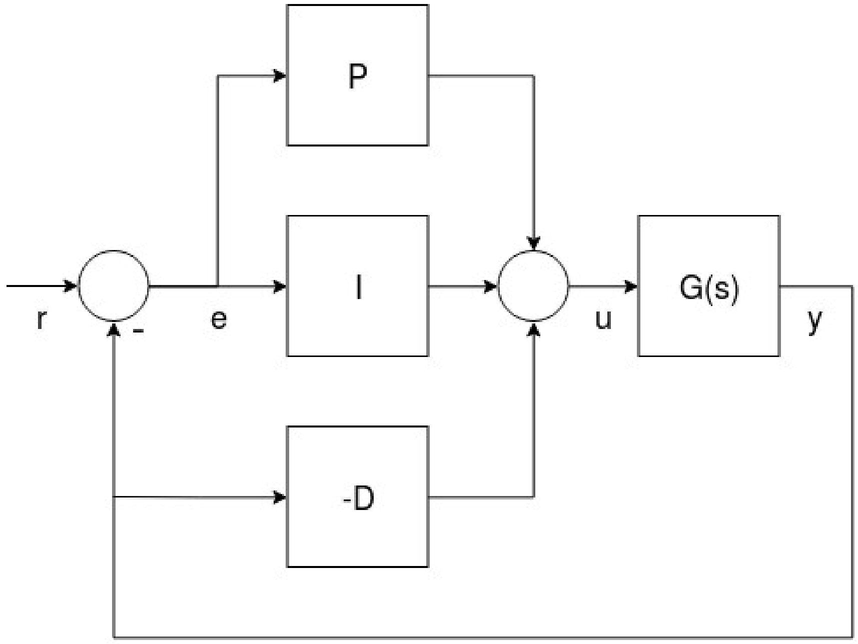


Figure A.8: Parallel Attitude Rate Px4 Control Scheme

This mode is recommended to be used when tuning is complete as the output is the sum of the P, I, and D gains. This is the most common form described in control systems literature and more intuitive for human understanding.

The final output of the angular rate controller are the thrust setpoints which are converted by the mixer into the appropriate angular velocities for each rotor. The controller can be tuned for radio-controlled flying which makes manual maneuvers easier but in autonomy mode, the controller is highly stable and adept at disturbance rejection. The quality of the control system will be judged in the results section when position setpoints will be compared against measured locations.

APPENDIX B

Start and End Points of A* Paths

	Latitude	Longitude
Start1	43.1743947663923	-70.9309812347388
End1	43.1748177518977	-70.9309945669665
Start2	43.1748177518977	-70.9309945669665
End2	43.1748156757165	-70.9311175331589
Start3	43.1748156757165	-70.9311175331589
End3	43.1744984026096	-70.9312428663477
Start4	43.1744984026096	-70.9312428663477
End4	43.1746716806651	-70.9311129942072
Start5	43.1746716806651	-70.9311129942072
End5	43.1748575596450	-70.9313034009875
Start6	43.1748575596450	-70.9313034009875
End6	43.1744538194279	-70.9312168547551

Table B.1: UAV 1 Start and End Point Coordinates of A* Paths

	Latitude	Longitude
Start1	43.1743962196191	-70.9308951589898
End1	43.1748039044787	-70.9307480672798
Start2	43.1748039044787	-70.9307480672798
End2	43.1746994369071	-70.9305356215430
Start3	43.1746994369071	-70.9305356215430
End3	43.1743206195046	-70.9305728965974
Start4	43.1743206195046	-70.9305728965974
End4	43.1745501779866	-70.9303094621202

Table B.2: UAV 3 Start and End Point Coordinates of A* Paths

	Latitude	Longitude
Start1	43.17433198	-70.93096695
End1	43.17382758	-70.93097566
Start2	43.17382758	-70.93097566
End2	43.17384329	-70.93111149
Start3	43.17384329	-70.93111149
End3	43.17423886	-70.93114856
Start4	43.17423886	-70.93114856
End4	43.17418971	-70.93139307
Start5	43.17418971	-70.93139307
End5	43.1738435	-70.93109919

Table B.3: UAV 3 Start and End Point Coordinates of A* Paths

APPENDIX C

Supplementary Parking Lot GPS Comparison

For each of the seven test flights, the maximum difference between the Xsens GPS and the navigational GPS is calculated for each flight. The navigational GPS has a maximum measurement rate of 10 Hertz and the Xsens GPS has a maximum measurement rate of 400 Hertz. Therefore, for the same flight the Xsens has more data points compared to that of the Pixhawk GPS. To calculate the difference between the two, the Pixhawk GPS measurements are linearly interpolated to extract GPS data that is compatible and directly comparable to that of the Xsens GPS data.

The measurement difference is shown in Table 5.1.

	Maximum (m)	Average (m)	Mean Average Deviation (m)	RMSE (m)
Test1	5.99	3.41	0.300	1.00
Test2	6.40	1.90	0.669	2.15
Test3	5.29	4.69	0.088	4.77
Test4	5.53	3.75	0.437	3.89
Test5	5.88	2.26	0.702	2.45
Test6	4.00	1.52	0.580	1.65
Test7	4.32	1.93	0.828	2.06

Table C.1: Xsens and Pixhawk 4 GPS Measurement Comparison Statistics